

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КОНОТОПСЬКИЙ ІНСТИТУТ

Кафедра електронних  
приладів і автоматики

Дипломна робота

**Розрахунок кількості інформації на символ повідомлення в  
інформаційних системах передачі**

студента гр. ЕПз-61к

В.В.Закорський

Науковий керівник,

ст. викладач, к.т.н.

Нормоконтроль,

---

Ю.В.Столярчук

---

Конотоп 2020

## РЕФЕРАТ

Об'єктом дослідження кваліфікаційної роботи є програма Turbo Pascal.

Мета роботи полягає у розрахуванні кількості інформації про символні повідомлення, що складаються з алфавіту, стану з 24 символів, для двох випадків: якщо символи алфавіту зустрічаються з рівними ймовірностями та якщо вірогідність нерівна.

У даній кваліфікаційній роботі було проведено розрахунок початкових ймовірностей для нерівноймовірних символів алфавіту, було виконано нормування зазначених ймовірностей, розраховано ентропію алфавіту з рівноймовірних символів, обчислено швидкість передачі та надмірність, побудовано оптимальний код за методом Шеннона-Фано.

У результаті проведених досліджень встановлено, що кількість інформації на символ для рівно ймовірного алфавіту - 4,585 біт / сим, кількість інформації на символ для нерівно ймовірного алфавіту - 2,6409 біт / сим, недовантаження символів - 1,9441 біт / сим, швидкість передачі інформації - 0,1244 біт / сек, надмірність повідомлення - 0,4240, побудований наступний оптимальний код:

Робота викладена на 38 сторінках, у тому числі включає 3 таблиці та список цитованої літератури із 41 джерела.

**КЛЮЧОВІ СЛОВА:** ЕНТРОПІЯ АЛФАВІТУ, ШВИДКІСТЬ ПЕРЕДАЧІ, НАДМІРНІСТЬ, МЕТОД ШЕННОНА-ФАНО, НАДМІРНІСТЬ ПОВІДОМЛЕННЯ.

## ЗМІСТ

	<b>с.</b>
<b>ВСТУП</b> .....	4
<b>ЗМІСТ ЗАВДАННЯ</b> .....	5
<b>ТЕОРЕТИЧНА ЧАСТИНА</b> .....	7
КІЛЬКІСНА ОЦІНКА ІНФОРМАЦІ.....	7
ОБЧИСЛЕННЯ ШВИДКОСТІ ПЕРЕДАЧІ ІНФОРМАЦІЇ І ПРОПУСКНОЇ СПРОМОЖНОСТІ КАНАЛІВ ЗВ'ЯЗКУ.....	9
ВИЗНАЧЕННЯ НАДЛИШКОВОГО ПОВІДОМЛЕНЬ ОПТИМАЛЬНИЙ КОДУВАННЯ.....	13
<b>ПРАКТИЧНА ЧАСТИНА</b> .....	17
РОЗРАХУНКИ.....	17
ВИЗНАЧЕННЯ КІЛЬКОСТІ ІНФОРМАЦІЇ НА СИМВОЛ ПОВІДОМЛЕННЯ. ПОБУДОВА ОПТИМАЛЬНОГО КОДУ.....	23
ТЕКСТ ПРОГРАМИ.....	26
<b>ВИСНОВОК</b> .....	33
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	35

## ВСТУП

Інформатика та обчислювальна техніка - це область науки і техніки, яка включає сукупність засобів, способів і методів людської діяльності, спрямованих на створення і застосування пристроїв зв'язку, систем збору, зберігання і обробки інформації.

У багатьох випадках зберігається та передається інформація може становити інтерес для осіб, які бажають використовувати її в корисливих цілях.

Одним з методів захисту є кодування. Кодування - це відображення повідомлень кодом за певним правилом присвоєння символів.

Код - це правило, яке описує відображення одного набору знаків в інший набір знаків (або слів). Кодом також називають і безліч образів при цьому відображенні.

Оптимальний код - це найбільш ефективний випадок кодування з нульовою надмірністю. При усуненні надмірності істотно знижується кількість символів, необхідних для кодованих повідомлень. Внаслідок цього зменшується час передачі, знижується необхідний обсяг пам'яті.

Таким чином, знання методів обробки інформації є базовим для інженерів, робота яких пов'язана з обчислювальними системами і мережами. Надмірність - додаткові кошти, що вводяться в систему для підвищення її надійності і захищеності.

Таким чином, інформатика займається вивченням обробки і передачі інформації.

В роботі відбивається застосування базових понять інформатики.

## ЗМІСТ ЗАВДАННЯ

Для проведення розрахунків розробити програму на мові ПАСКАЛЬ.

1.1. Число символів алфавіту  $k = m$  (номер варіанта завдання) + 10.  
Визначити кількість інформації на символ повідомлення, складеного з цього алфавіту:

- а) якщо символи алфавіту зустрічаються з рівними можливостями;
- б) якщо символи алфавіту зустрічаються в повідомленні з можливостями:

$$p_1 = 0,15; p_2 = p_1/(k-1); p_3 = (p_1 + p_2)/(k-2) \dots$$

$$k-1$$

$$p_k = \sum_{n=1} p_n / (k - k + 1).$$

$$n=1$$

Сума всіх ймовірностей повинна бути рівною одиниці, тому:

$$p_i$$

$$p_i = \frac{p_i}{\sum_{j=1}^k p_j}$$

$$k$$

$$\sum p_j$$

$$j=1$$

Визначити, наскільки недовантажені символи в другій ситуації.

1.2. Число символів алфавіту =  $m$  (номер варіанта завдання).  
Ймовірності появи символів рівні відповідно

$$p_1 = 0,15; p_2 = p_1/(k-1); p_3 = (p_1 + p_2)/(k-2) \dots$$

$k-1$

$$p_k = \sum_{n=1}^{k-1} p_n / (k - k + 1).$$

$n=1$

Тривалості символів  $\tau_1 = 1$  сек;  $\tau_2 = 2$  сек;

$$\tau_k = \tau_{k-1} + 1.$$

Чому дорівнює швидкість передачі повідомлень, складених з таких символів?

Визначити кількість інформації на символ повідомлення, складеного з цього алфавіту:

а) якщо символи алфавіту зустрічаються з рівними можливостями;

Визначити, наскільки недовантажені символи в другому випадку.

1.3. Повідомлення складаються з алфавіту з числом символів  $= m$ .

Імовірність появи символів алфавіту дорівнює відповідно:

$$p_1 = 0,15; p_2 = p_1/(k-1); p_3 = (p_1 + p_2)/(k-2) \dots$$

$k-1$

$$p_k = \sum_{n=1}^{k-1} p_n / (k - k + 1).$$

$n=1$

Знайти надмірність повідомлень, складених з даного алфавіту.

Побудувати оптимальний код повідомлення.

## ТЕОРЕТИЧНА ЧАСТИНА

### КІЛЬКІСНА ОЦІНКА ІНФОРМАЦІЇ

Загальна кількість неповторюваних повідомлень, яке може бути складено з алфавіту  $m$  шляхом комбінування по  $n$  символів в повідомленні,

$$N = m^n$$

Невизначеність, яка припадає на символ первинного (кодованого) алфавіту, складеного з рівноймовірно і взаємoneзалежних символів,

$$H = \log_2 m$$

Так як інформація є невизначеність, що знімається при отриманні повідомлення, то кількість інформації може бути представлено як твір загальної кількості повідомлень  $k$  на середню ентропію  $H$ , що припадає на одне повідомлення:

$$I = k * H \text{ біт}$$

Для випадків рівно ймовірних і взаємoneзалежних символів первинного алфавіту кількість інформації в  $k$  повідомленнях алфавіту  $m$  одно:

$$I = k * \log_2 m \text{ біт}$$

Для нерівно ймовірних алфавітів ентропія на символ алфавіту:

$m$

$$H = \sum_{i=1}^m p_i * \log_2(1/p_i) = -\sum_{i=1}^m p_i * \log_2 p_i \text{ біт/символ}$$

$$i=1 \quad i=1$$

А кількість інформації в повідомленні, складеному з  $k$  не рівновероятних символів,

$m$

$$I = -k \sum_{i=1} p_i \log_2 p_i \text{ біт}$$

$$i=1$$



## ОБЧИСЛЕННЯ ШВИДКОСТІ ПЕРЕДАЧІ ІНФОРМАЦІЇ І ПРОПУСКНОЇ СПРОМОЖНОСТІ КАНАЛІВ ЗВ'ЯЗКУ

В умовах відсутності перешкод швидкість передачі інформації визначається кількістю інформації, пореносним символом повідомлення в одиницю часу, і дорівнює

$$C = n * H,$$

де  $n$  - кількість символів, що виробляються джерелом повідомлень за одиницю часу;  $H$  - ентропія (невизначеність), що знімається при отриманні одного символу повідомлень, що виробляються даними джерелом.

Швидкість передачі інформації також може бути представлена як

$$C = \frac{H}{\tau} \text{ біт/сек,}$$

де  $\tau$  - час передачі одного двійкового символу.

Для повідомлень, складених з рівно ймовірно взаємонезалежних символів рівної тривалості, швидкість передачі інформації:

$$C = (1/\tau) * \log_2 m \text{ біт/сек}$$

У разі неравновероятних символів рівної тривалості:

$m$

$$C = (1/\tau) * \sum_{i=1}^m p_i * \log_2 p_i \text{ біт/сек}$$

$i=1$

У разі неавновероятних і взаємозалежних символів різної тривалості:

$$C = \frac{- \sum_{i=1}^m p_i \cdot \log_2 p_i}{\sum_{i=1}^m \tau_i \cdot p_i}$$

Пропускна здатність (або ємність каналу зв'язку) - є максимальна швидкість передачі інформації з даного каналу зв'язку. Під каналом зв'язку мається на увазі сукупність засобів, призначених для передачі інформації від даного джерела повідомлень до адресата. Вираз для пропускної здатності відрізняється тим, що пропускну здатність характеризує максимальна ентропія:

$$C_{\text{макс}} = \frac{H_{\text{макс}}}{\tau} \text{ біт/сек}$$

Для двійкового коду:

$$C_{\text{макс}} = \frac{\log_2 2}{\tau} = \frac{1}{\tau} \text{ біт/сек}$$

При наявності перешкод пропускна здатність каналу зв'язку обчислюється як добуток кількості прийнятих в секунду знаків  $n$  на різницю ентропії джерела повідомлень і умовної ентропії джерела повідомлень щодо прийнятого сигналу:

$$C_{II} = n \cdot [H(A) - H(A/B)] \text{ біт/сек (15)}$$

або

$$C_{\Pi} = -n \cdot \left[ \sum_i p_i \cdot \log_2 p_i + \sum_i \sum_j p(a_i) \cdot p(b_j / a_i) \cdot \log_2 p(b_j / a_i) \right] \text{ біт/сек}$$

У загальному випадку

$$\begin{aligned} C_{\Pi} &= n \cdot [H(A) - H(A/B)] = n \cdot [H(B) - H(B/A)] = \\ &= n \cdot [H(A) + H(B) - H(B, A)] \text{ біт/сек (16)} \end{aligned}$$

Якщо символи джерела повідомлень нерівно ймовірні і взаємно залежні, то ентропія джерела обчислюється за формулою загальної умовної ентропії.

Для симетричних бінарних каналів, в яких сигнали передаються за допомогою двох якісних ознак і ймовірність помилкового прийому  $p_{\Pi} = p(1/0) = p(0/1)$ , а ймовірність правильного прийому  $p_{\Pi} = 1 - p_{\Pi}$ , втрати враховуються при допомозі умовної ентропії виду

$$H(B/A) = -[p_{\Pi} \cdot \log_2 p_{\Pi} + (1 - p_{\Pi}) \cdot \log_2 (1 - p_{\Pi})] \text{ біт/сек (17)}$$

пропускна здатність таких каналів

$$C_{\Pi} = n \cdot [1 + p_{\Pi} \cdot \log_2 p_{\Pi} + (1 - p_{\Pi}) \cdot \log_2 (1 - p_{\Pi})] \text{ біт/сек (18)}$$

Для симетричного бінарного каналу

$$\begin{aligned} C_{\Pi} &= n \cdot \{ p(A_1) \cdot [p(1/1) \cdot \log_2 p(1/1) + p(0/1) \cdot \log_2 p(0/1)] + \\ &+ p(A_0) \cdot [p(1/0) \cdot \log_2 p(1/0) + p(0/0) \cdot \log_2 p(0/0)] - \\ &- [p(B_0) \cdot \log_2 p(B_0) + p(B_1) \cdot \log_2 p(B_1)] \} \text{ біт/сек (19)} \end{aligned}$$

Для симетричних дискретних каналів зв'язку з числом якісних ознак  $m > 2$  пропускна здатність

$$C_{II} = n \cdot [\log_2 m + p_{II} \cdot \log_2 \frac{p_{II}}{m-1} + (1 - p_{II})] \text{ біт/сек (20)}$$

## ВИЗНАЧЕННЯ НАДЛИШКОВОГО ПОВІДОМЛЕННЯ. ОПТИМАЛЬНИЙ КОДУВАННЯ

Якщо ентропія джерела повідомлень не дорівнює максимальній ентропії для алфавіту з даними кількістю якісних ознак (маються на увазі якісні ознаки алфавіту, за допомогою яких складаються повідомлення), то це, перш за все, означає, що повідомлення даного джерела могли б нести більшу кількість інформації. Абсолютна недовантаження на символ повідомлень такого джерела:

$$\Delta D = (H_{\text{макс}} - H) \text{ біт/символ}$$

Для визначення кількості "зайвої" інформації, яка закладена в структурі алфавіту або в природі коду, вводиться поняття надмірності. Надмірність, з якою ми маємо справу в теорії інформації, не залежить від змісту повідомлення і зазвичай заздалегідь відома зі статистичних даних. Інформаційна надмірність показує відносну недовантаження на символ алфавіту і є безрозмірною величиною:

$$D = \frac{H_{\text{макс}} - H}{H_{\text{макс}}} = 1 - \frac{H}{H_{\text{макс}}},$$

де  $\frac{H}{H_{\text{макс}}} = \mu$  - коефіцієнт стиснення (відносна ентропія).  $H$  і  $H_{\text{макс}}$  беруться щодо одного і того ж алфавіту.

Крім загального поняття надмірності існують приватні види надмірності (надмірність, обумовлена нерівно ймовірним розподілом символів в повідомленні, надмірність, викликана статистичної зв'язком між символами повідомлення).

Надмірність, яка закладена в природі даного коду, виходить в результаті нерівномірного розподілу в повідомленнях якісних ознак цього коду і не може бути задана однією цифрою на підставі статистичних випробувань. Так при передачі десяткових цифр двійковим кодом максимально завантаженими бувають тільки ті символи вторинного алфавіту, які передають значення, що є цілочисельними ступенями двійки. В інших випадках тією ж кількістю символів може бути передано більшу кількість цифр (повідомлень). Наприклад, трьома двійковими розрядами ми можемо передати і цифру 5, і цифру 8. Фактично для передачі повідомлення достатньо мати довжину кодової комбінації.

Фактично для передачі повідомлення достатньо мати довжину кодової комбінації

$$L \geq \frac{\log_2 N}{\log_2 m}$$

де  $N$  - загальна кількість переданих повідомлень.

$L$  можна представити і як

$$L \geq \frac{\log_2 m_1}{\log_2 m_2}$$

де  $m_1$  і  $m_2$  - відповідно якісні ознаки первинного та вторинного алфавітів. Тому для цифри 5 у двійковому кодi можна записати

$$L \geq \frac{\log_2 5}{\log_2 2} = 2,32$$

дв. симв.

Однак цю цифру необхідно округлити до найближчого цілого числа (в більшу сторону), так як довжина коду не може бути виражена дробовим числом.

У загальному випадку, надмірність від округлення:

$$D_0 = \frac{k - \varphi}{k}$$

де  $\varphi = \frac{\log_2 m_1}{\log_2 m_2}$ ,  $k$  - округлене до найближчого цілого числа значення.

Для нашого прикладу

$$D_0 = \frac{3 - 2,32}{3} \approx 0,227$$

Надмірність необхідна для підвищення завадостійкості кодів і її вводять штучно у вигляді додаткових  $n_k$  символів. Якщо в коді всього  $n$  розрядів і  $n_{II}$  з них несуть інформаційне навантаження, то  $n_k = n - n_{II}$  характеризують абсолютну коригувальну надмірність, а величина

$$D_k = \frac{n - n_{II}}{n_{II}}$$

характеризує відносну коригувальну надмірність.

Для зменшення надмірності використовують оптимальні коди. При побудові оптимальних кодів найбільшого поширення набули методики Шеннона-Фано і Хаффмена. Згідно з методикою Шеннона-Фано побудова оптимального коду ансамблю з повідомлень зводиться до наступного:

1) безліч з повідомлень розташовується в порядку убутання ймовірностей;

2) початковий ансамбль кодованих сигналів розбивається на дві групи таким чином, щоб сумарні ймовірності повідомлень обох груп були по можливості рівні.

Якщо рівній ймовірності в підгрупах не можна досягти, то їх ділять так, щоб у верхній частині (верхньої підгрупі) залишалися символи, сумарна ймовірність яких менше сумарної ймовірності символів у нижній частині (нижньої підгрупі);

3) першій групі присвоюється символ 0, а другій групі - символ 1;

4) кожен з утворених підгруп ділять на дві частини таким чином, щоб сумарні ймовірності новостворених підгруп були по можливості рівні;

5) першим групам кожної з підгруп знову присвоюється 0, а другим - 1. Таким чином, ми отримуємо другі цифри коду. Потім кожна з чотирьох груп знову ділиться на рівні (з точки зору сумарної ймовірності) частини до тих пір, поки в кожній з підгруп не залишиться по одній букві.

Побудований код називають оптимальним нерівномірним кодом (ОНК).



## ПРАКТИЧНА ЧАСТИНА

### РОЗРАХУНКИ

- 1) розраховується початкові ймовірності для нерівно ймовірних символів алфавіту.
- 2) виконує нормування зазначених ймовірностей.
- 3) розраховується ентропія алфавіту з рівно ймовірних символів.
- 4) проводиться розрахунок ентропії алфавіту з нерівно ймовірними символами і недовантаження в цьому випадку.
- 5) з урахуванням заданих тривалостей символів обчислюється швидкість передачі і надмірність.
- б) будується оптимальний код за методом Шеннона-Фано.

Розрахунок ймовірностей.

Проміжні значення:	Остаточний результат:
$k-1$ $\dots p_k = S p_n / (m - k + 1).$ $n-1$	$p_i = p_i / \left( \sum_{i=1}^k p_i \right)$
$p_1 = 0,1500$ $p_2 = 0,0065$ $p_3 = 0,0071$ $p_4 = 0,0078$ $p_5 = 0,0086$ $p_6 = 0,0095$ $p_7 = 0,0105$ $p_8 = 0,0118$ $p_9 = 0,0132$ $p_{10} = 0,0150$ $p_{11} = 0,0171$ $p_{12} = 0,0198$ $p_{13} = 0,0231$ $p_{14} = 0,0273$ $p_{15} = 0,0327$ $p_{16} = 0,0400$	$p_1=0,0417$ $p_2=0,0018$ $p_3=0,0020$ $p_4=0,0022$ $p_5=0,0024$ $p_6=0,0026$ $p_7=0,0029$ $p_8=0,0033$ $p_9=0,0037$ $p_{10}=0,0042$ $p_{11}=0,0048$ $p_{12}=0,0055$ $p_{13}=0,0064$ $p_{14}=0,0076$ $p_{15}=0,0091$ $p_{16}=0,0111$

$p_{17} = 0,0500$	$p_{17} = 0,0139$
$p_{18} = 0,0643$	$p_{18} = 0,0179$
$p_{19} = 0,0857$	$p_{19} = 0,0238$
$p_{20} = 0,1200$	$p_{20} = 0,0333$
$p_{21} = 0,1800$	$p_{21} = 0,0500$
$p_{22} = 0,3000$	$p_{22} = 0,0833$
$p_{23} = 0,6000$	$p_{23} = 0,1667$
$p_{24} = 1,8000$	$p_{24} = 0,5000$
$\sum_{i=1}^k p_i = 3,6$	$\sum_{i=1}^k p_i = 1$

Визначення кількості інформації на символ повідомлення, складеного з даного алфавіту.

Кількість інформації на символ повідомлення для символів даного алфавіту, що зустрічаються з рівними можливостями:

$$H_{\max} = \log_2 24 = \ln 24 / \ln 2 = 4,5850 \text{ біт/символ}$$

Кількість інформації на символ повідомлення для символів даного алфавіту, що зустрічаються в повідомленні з різними можливостями:

$$\begin{aligned}
H = & - (0,0417 * \log_2 0,0417 + 0,0018 * \log_2 0,0018 + 0,020 * \log_2 0,0020 + \\
& 0,0022 * \log_2 0,0022 + 0,0024 * \log_2 0,0024 + 0,0026 * \log_2 0,0026 + \\
& 0,0029 * \log_2 0,0029 + 0,0033 * \log_2 0,0033 + 0,0037 * \log_2 0,0037 + \\
& 0,0042 * \log_2 0,0042 + 0,0048 * \log_2 0,0048 + 0,0055 * \log_2 0,0055 + \\
& 0,0064 * \log_2 0,0064 + 0,0076 * \log_2 0,0076 + 0,0091 * \log_2 0,0091 + \\
& 0,0111 * \log_2 0,0111 + 0,0139 * \log_2 0,0139 + 0,0179 * \log_2 0,0179 + \\
& 0,0238 * \log_2 0,0238 + 0,0333 * \log_2 0,0333 + 0,0500 * \log_2 0,0500 + \\
& 0,0833 * \log_2 0,0833 + 0,1667 * \log_2 0,1667 + 0,5000 * \log_2 0,5000) = \\
& = 2,6409 \text{ біт/символ}
\end{aligned}$$

Недовантаження символів в даному випадку:

$$N = H_{\max} - H = 4,5850 - 2,6409 = 1,9441 \text{ біт/символ}$$

Обчислення швидкості передачі інформації.

$$\begin{aligned} C = & - (0,0417 * \log_2 0,0417 + 0,0018 * \log_2 0,0018 + 0,020 * \log_2 0,0020 + \\ & 0,0022 * \log_2 0,0022 + 0,0024 * \log_2 0,0024 + 0,0026 * \log_2 0,0026 + \\ & 0,0029 * \log_2 0,0029 + 0,0033 * \log_2 0,0033 + 0,0037 * \log_2 0,0037 + \\ & 0,0042 * \log_2 0,0042 + 0,0048 * \log_2 0,0048 + 0,0055 * \log_2 0,0055 + \\ & 0,0064 * \log_2 0,0064 + 0,0076 * \log_2 0,0076 + 0,0091 * \log_2 0,0091 + \\ & 0,0111 * \log_2 0,0111 + 0,0139 * \log_2 0,0139 + 0,0179 * \log_2 0,0179 + \\ & 0,0238 * \log_2 0,0238 + 0,0333 * \log_2 0,0333 + 0,0500 * \log_2 0,0500 + \\ & 0,0833 * \log_2 0,0833 + 0,1667 * \log_2 0,1667 + 0,5000 * \log_2 0,5000) / \\ & (1 * 0,0417 + 2 * 0,0018 + 3 * 0,020 + 4 * 0,0022 + 5 * 0,0024 + 6 * 0,0026 + \\ & 7 * 0,0029 + 8 * 0,0033 + 9 * 0,0037 + 10 * 0,0042 + 11 * 0,0048 + 12 * 0,0055 + \\ & 13 * 0,0064 + 14 * 0,0076 + 15 * 0,0091 + 16 * 0,0111 + 17 * 0,0139 + 18 * 0,0179 + \\ & 19 * 0,0238 + 20 * 0,0333 + 21 * 0,0500 + 22 * 0,0833 + 23 * 0,1667 + 24 * 0,5000) = \\ & 0,1244 \text{ біт/сек} \end{aligned}$$

Надмірність повідомлень, складених з даного алфавіту.

$$D = 1 - (H/H_{\max}) = 1 - (2,6409 / 4,5850) = 0,4240$$



22	p4=0,0022	1	0	1	0	1
23	p3=0,0020	1	0	1	0	1
24	p2=0,0018	1	0	1	0	1

0		0,0022	0			10101000
0		0,0038	1	0,0020	1	101010011
0	0,0083	0	1	0,0018	0	101010010

Буква	Імовірність появи букви	Кодові слова	Число знаків в кодовому слові	$P_i \cdot l_i$
A[1] (p24)	0,5000	0	1	0,5
A[2] (p23)	0,1667	111	3	0,50001
A[3] (p22)	0,0833	110	3	0,2500
A[4] (p21)	0,0500	1000	4	0,2000
A[5] (p 1)	0,0417	10011	5	0,2083
A[6] (p20)	0,0333	10010	5	0,1667
A[7] (p19)	0,0238	101111	6	0,1429
A[8] (p18)	0,0179	1011100	7	0,1250
A[9] (p17)	0,0139	101101	6	0,0833
A[10] (p16)	0,0111	101110	6	0,0667
A[11] (p15)	0,0091	1010011	7	0,0636
A[12] (p14)	0,0076	10100100	8	0,0606
A[13] (p13)	0,0064	1010001	7	0,0449
A[14] (p12)	0,0055	1010011	7	0,0385
A[15] (p11)	0,0048	10101111	8	0,0381
A[16] (p10)	0,0042	101011100	9	0,0375
A[17] (p9)	0,0037	10101101	8	0,0294
A[18] (p8)	0,0033	10101110	8	0,0261
A[19] (p7)	0,0029	10101010	8	0,0234
A[20] (p6)	0,0026	101010111	9	0,0237
A[21] (p5)	0,0024	101010110	9	0,0214
A[22] (p4)	0,0022	10101000	8	0,0173
A[23] (p3)	0,0020	101010011	9	0,0178
A[24] (p2)	0,0018	101010010	9	0,0163

ВИЗНАЧЕННЯ КІЛЬКОСТІ ІНФОРМАЦІЇ НА СИМВОЛ  
ПОВІДОМЛЕННЯ.  
ПОБУДОВА ОПТИМАЛЬНОГО КОДУ

З початок безліч з повідомлень розташуємо в порядку убутання ймовірностей. Потім, розіб'ємо дане безліч на дві групи таким чином, щоб сумарні ймовірності повідомлень обох груп були по можливості рівні. Але оскільки рівність не досягається, то ми їх ділимо так, щоб у верхній частині залишалися символи, сумарна ймовірність яких менше сумарної ймовірності символів у нижній частині. Першій групі присвоюємо символ 0, а другий групі = символ 1. кожен з утворених підгруп ділимо на дві частини таким чином, щоб сумарні ймовірності новостворених підгруп були по можливості рівні. Першим групам кожної з підгруп знову присвоюємо 0, а другим 1. таким образом ми отримуємо ми отримуємо другі цифри коду. Потім кожен з чотирьох груп знову ділимо на рівні частини до тих пір, поки в кожній з підгруп не залишиться по одній букві.

Оптимальний код (вийшов результат):

Буква	Ймовірність появи букви	Кодове слово	Число знаків в кодовому слові	$p_i \cdot l_i$
P1	0,055	000	3	0,165
P2	0,053	0010	4	0,212
P3	0,051	00110	5	0,255
P4	0,050	00111	5	0,250
P5	0,048	0100	4	0,192
P6	0,046	0101	4	0,176
P7	0,044	0110	4	0,114
P8	0,043	01110	5	0,215
P9	0,041	011110	6	0,246
P10	0,040	011111	6	0,240
P11	0,039	1000	4	0,156
P12	0,038	10010	5	0,190
P13	0,036	10011	5	0,180

P14	0,035	1010	4	0,140
P15	0,033	10110	5	0,165
P16	0,032	101110	6	0,192
P17	0,030	101111	6	0,180
P18	0,029	11000	5	0,145
P19	0,027	11001	5	0,135
P20	0,026	11010	5	0,130
P21	0,025	110110	6	0,150
P22	0,023	110111	6	0,138
P23	0,022	11100	5	0,110
P24	0,020	111010	6	0,120
P25	0,019	111011	6	0,114
P26	0,018	111100	6	0,108
P27	0,017	111101	6	0,102
P28	0,016	111110	6	0,096
P29	0,013	1111110	7	0,091
P30	0,012	11111110	8	0,096
P31	0,010	11111111	8	0,080

Ручне побудова ОНК за методикою Шеннона-Фано:

P1	0,010	11111111				0,022	0,010
P2	0,012	11111110			0,051	0,035	0,012
P3	0,013	1111110			0,086		0,013
P4	0,016	111110				0,016	
P5	0,017	111101		0,147		0,017	
P6	0,018	111100				0,035	0,018
P7	0,019	111011		0,277		0,039	0,019
P8	0,020	111010			0,061		0,020
P9	0,022	11100				0,022	
P10	0,023	110111		0,520			0,023
P11	0,025	110110			0,074	0,048	0,025
P12	0,026	11010			0,130	0,026	
P13	0,027	11001				0,027	
P14	0,029	11000				0,056	0,029
P15	0,030	101111					0,030
P16	0,032	101110				0,062	0,032
P17	0,033	10110			0,130		
P18	0,035	1010		0,243		0,033	
P19	0,036	10011				0,035	
P20	0,038	10010			0,113	0,074	0,036
						0,038	



P21	0,039	1000			0,039		
P22	0,040	011111				0,081	0,040
P23	0,041	011110			0,124		0,041
P24	0,043	01110		0,168		0,043	
P25	0,044	0110		0,262	0,044		
P26	0,046	0101	0,471		0,046		
P27	0,048	0100			0,094		
P28	0,050	00111			0,048		
P29	0,051	00110				0,101	0,050
P30	0,053	0010		0,209	0,154		0,051
P31	0,055	000			0,053		
					0,055		

## ТЕКСТ ПРОГРАМИ

```
uses Crt,Graph;
const k=24;
type
  mass=array [1..k] of real;
var
  i,x: integer;
  s,H,Hmax,deltaD,sum,C,sumTiPi,D: real;
  p,a: mass;
  code: array [1..k] of string[20];

{Процедура побудови оптимального коду по методике Шеннона-Фано}
procedure shannona(b:mass);
  procedure divide(var nS:integer; n1,n2:integer);
    var
      s,s1,s2: real;
    begin
      s:=0;
      for i:=n1 to n2 do s:=s+a[i];
      s1:=0; s2:=0;
      i:=n1-1;
      repeat
        inc(i);
        s1:=s1+a[i];
        s2:=s1+a[i+1];
      until abs(s/2-s1)<abs(s/2-s2);
      nS:=i;
      for x:=n1 to nS do
        if (s/2-s1)>=0 then code[x]:=code[x]+'0'
```

```

else code[x]:=code[x]+'1';
for x:=nS+1 to n2 do
if (s/2-s1)<0 then code[x]:=code[x]+'0'
else code[x]:=code[x]+'1';
end;
var
tmp: real;
j,n1,n2,nS: integer;
begin
for i:=1 to k do code[i]:= "";
for i:=1 to k do a[i]:=b[i];
for i:=1 to k do
for j:=k downto(i+1) do
if a[i]<a[j]
then
begin
tmp:=a[i];
a[i]:=a[j];
a[j]:=tmp;
end;
j:=1;
repeat
divide(nS,j,k);
n1:=nS;
while (nS-j)>0 do divide(nS,j,nS);
j:=nS+1;
n2:=n1;
while (n1-j)>0 do divide(n1,j,n1);
j:=n2+1;
until j>(k-1);

```

```

end;
procedure zastavka;
var dr,reg,err:integer;
begin
dr:=detect;reg:=detect;
initgraph(dr,reg,'d:\tp7\tpu\');
err:=graphresult;
if err<>grok then
begin
writeln('Ошибка ініціалізації графічного модуля!');
halt;
end;
setcolor(19);
settextstyle(3,0,4);
outtextxy(150,40,'Расчетно-графическая работа');
outtextxy(240,65,'на тему');
setcolor(14);
settextstyle(4,0,4);
outtextxy(50,125,'"Построение простого коду методом Шеннона-
Фано"');
settextstyle(6,0,2);
setcolor(19);
outtextxy(325,250,'Выполнил:');
settextstyle(6,0,2);
setcolor(10);
outtextxy(400,250,'Калинин С.А. ПС-11');
outtextxy(200,450,'Натисніть любую клавішу');
readln;
closegraph;
end;

```

```

procedure vivod;
begin
textcolor(lightgreen);
writeln('Оптимальний код: '); {вивод кінцевої таблиці}
writeln('Символ':7,'Ймовірність':13,'Оптимальний код':20,'Число
зн.':15,'Ймовір.*Числ.зн.':20);
for i:=1 to k do
begin
write(' p[,i:2,] ');
write(p[i]:0:4,' ');
write(code[i]:20,' ');
write(length(code[i]):15,' ');
write((p[i]*length(code[i])):0:4);
if i<>k then writeln;
end;
end;
begin
zastavka;
clrscr;
{1.1 а) Кількість інформації про символ повідомлень,
составленого з алфавіту рівно ймовірних символів }
Hmax:=ln(k)/ln(2);
{1.1 б) Розрахунок ймовірності для нерівноймовірних символів }
p[1]:=0.15;
sum:=p[1];
for i:=2 to k do
begin
p[i]:=sum/(k+1-i);
sum:=sum+p[i];
end;

```

```

clrscr;
textcolor(11);
writeln('Промежні значення ймовірності: ');
writeln;
textcolor(10);
for i:=1 to 14 do
writeln('Ймовірність p['i:2,'] = ',p[i]:4:4);
readkey;
clrscr;
textcolor(11);
writeln('Промежні значення ймовірності: ');
writeln;
textcolor(10);
for i:=15 to k do
writeln('Ймовірність p['i:2,'] = ',p[i]:4:4);
writeln;
textcolor(11);
for i:=1 to k do s:=s+p[i];
writeln('Сумма ймовірностей = ',s:4:2);
readkey;
H:=0;
sumTiPi:=0;
for i:=1 to k do
begin
p[i]:=p[i]/sum;
{1.1 б) Розрахунок ентропії для алфавіта нерівно ймовірних символів}
H:=H+p[i]*(ln(1/p[i])/ln(2));
sumTiPi:=sumTiPi+i*p[i];
end;
clrscr;

```

```

textcolor(11);
writeln('Кінцеве значення: ');
writeln;
textcolor(10);
for i:=1 to 14 do
writeln('Ймовірність p['i:2,'] = ',p[i]:4:4);
readkey;
clrscr;
textcolor(11);
writeln('Кінцеве значення: ');
writeln;
textcolor(10);
for i:=15 to k do
writeln('Ймовірність p['i:2,'] = ',p[i]:4:4);
writeln;
textcolor(11);
s:=0;
for i:=1 to k do s:=s+p[i];
writeln('Сумма ймовірностей = ',s:4:2);
readkey;
{1.1 б) Визначення недовантаження символів}
deltaD:=Hmax-H;
{1.2 Розрахунок швидкості передачі повідомлення}
C:=H/SumTiPi;
{1.3 Розрахунок надмірності повідомлень}
D:=1-H/Hmax;
{Виклик процедури побудови оптимального коду}
shannona(p);
{Вивод результатів}
clrscr;

```

```

textcolor(11);
{ writeln('Кількість символів алфавіта = 'k,');}
writeln('1.1 Кількість інформації на символ повідомлення:');
writeln(' a) для алфавіту рівноймовірних символів: ');
textcolor(10); writeln(' Hmax =',Hmax:7:4,' біт/символ');
textcolor(11); writeln(' b) для алфавіта нерівноймовірних символів: ');
textcolor(10); writeln(' H =',H:7:4,' біт/символ');
textcolor(11); write(' Недовантаження:');
textcolor(10); writeln(' дельтаD =',deltaD:7:4,' біт/символ');
textcolor(11); writeln;
Writeln('1.2 Швидкість передачі інформації:');
textcolor(10); writeln(' C =',C:7:4,' біт/сек');
textcolor(11); writeln;
Writeln('1.3 Надлишковість повідомлень:');
textcolor(10); writeln(' D =',D:7:4);
writeln;
TextColor(11);
write('Натисніть будь яку клавішу для виводу таблиці результатів
побудови. ');
readkey;
clrScr;
vivot;
readkey;

```

end.



## ВИСНОВОК

У моїй курсовій роботі я використав теоретичний матеріал і розробив на мові (високому рівні) програму Turbo Pascal. Нам було розраховано кількість інформації про символні повідомлення, що складаються з алфавіту, стану з 24 символів, для двох випадків:

- 1] якщо символи алфавіту зустрічаються з рівними ймовірностями;
- 2] якщо вірогідність не рівна.

Так само я визначив кількість недовантажених символів у другому випадку, вивісив кількість інформації про символні повідомлення та швидко передав повідомлення, створені з таких символів, наші повідомлення про незрозумілість повідомлень, створені з даного алфавіту. Побудував оптимальний код повідомлення, застосовуючи методику Шеннона-Фано: за допомогою послідовного розподілу безлічі ймовірностей на групи за принципом рівності сум ймовірностей я склав у відповідність кожному символу найбільш оптимальну двійкову комбінацію. Таким чином, було отримано оптимальний двійковий код для алфавіту з 31 символу.

В результаті виконання роботи були отримані наступні результати:

- кількість інформації на символ для рівно ймовірного алфавіту - 4,585 біт / сим;
- кількість інформації на символ для нерівно ймовірного алфавіту - 2,6409 біт / сим;
- недовантаження символів - 1,9441 біт / сим;
- швидкість передачі інформації - 0,1244 біт / сек;
- надмірність повідомлення - 0,4240;
- побудований наступний оптимальний код:

Символ	Ймовірність появи	Код	Число знаків
p[ 1]	0.0417	0	
p[ 2]	0.0018	111	
p[ 3]	0.0020	110	
p[ 4]	0.0022	1000	
p[ 5]	0.0024	10011	
p[ 6]	0.0026	10010	
p[ 7]	0.0029	101111	
p[ 8]	0.0033	1011100	
p[ 9]	0.0037	101101	
p[10]	0.0042	101101	
p[11]	0.0048	1010011	
p[12]	0.0055	10100100	
p[13]	0.0064	1010001	
p[14]	0.0076	1010001	
p[15]	0.0091	10101111	
p[16]	0.0111	101011100	
p[17]	0,0139	10101101	
p[18]	0,0179	10101101	
p[19]	0,0238	10101010	
p[20]	0,0333	101010111	
p[21]	0,0500	101010110	
p[22]	0,0833	10101000	
p[23]	0,1667	101010011	
p[24]	0,5000	101010010	

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бауэр Ф. Информатика, М. 1992.
2. Колесник В.Д. Курс теории информации, М. 1982.
3. Фаронов В. В. Turbo Pascal 7.0. Учебное пособие, М. 2000.
4. Цымбаль В.П. Задачник по теории информации и кодированию, Киев. 1976.
5. Марченко А.И. Программирование в среде Turbo Pascal 7.0.
6. Горбоконеко В.Д., Шикина В.Е. Кодирование информации. Ульяновск: УлГТУ, 2006. – 56с.
7. Дмитриев, В. Н. Прикладная теория информации: учебник для вузов / В. Н. Дмитриев. – М.: Высшая школа, 1989. – 320 с.; ил.
8. Темников, Ф. Е. Теоретические основы информационной техники: учебное пособие для вузов / Ф. Е. Темников, В. А. Афонин, В. И. Дмитриев. – 2-е изд., перераб и доп. – М.: Энергия, 1979. – 512 с., ил.
9. Цымбал, В. П. Теория информации и кодирования: учебник для вузов / В. П. Цимбал. – 3-е изд., перераб. и доп. – Киев: Головное издательство «Вища школа», 1982. – 304 с.
10. Калабеков, Б. А. Микропроцессоры и их применение в системах передачи и обработки сигналов: учебное пособие для вузов / В. А. Калабеков. – М.: Радио и связь, 1988. – 368 с., ил.
11. Острейковский, В. А. Информатика: учебник для вузов / В. А. Острейковский. – М.: Высшая школа, 1991. – 551 с., ил.
12. Гиттис, Э. И. Преобразователи информации для электронных цифровых и вычислительных устройств. – 2-е изд., перераб. / Э. И. Гиттис. – М.: Энергия, 1970.

13. Гиттис, Э. И. Аналого-цифровые преобразователи: учебное пособие для вузов / Э. И. Гиттис, Е. А. Пискулов. – М.: Энергоиздат, 1981. – 380 с., ил.
14. М. Вернер. Основы кодирования. Учебник для ВУЗов. Москва: Техносфера, 2004. - 288с.
15. Хаффман Д.А. Метод построения кодов с минимальной избыточностью: Пер. с англ. //Кибернетический сборник. – М.: ИЛ, 1961. – Вып. 3. – С. 79–87;
16. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ–МИФИ, 2002;
17. Вернер М. Основы кодирования. Учебник для ВУЗов. – М.: Техносфера, 2004;
18. Лидовский В. В. Теория информации: Учебное пособие. — М.: Компания Спутник+, 2004.
19. Набережнов Г. М., Пьянов И. П., Чугунов Е. Н., Шлеймович М. П. Компьютерная графика: Учебное пособие для студентов специальности «Автоматизированные системы обработки информации и управления» /Под общ. ред. к. т. н., доцента Набережнова Г. М. – 2 – е изд., доп. – Казань: ИСПО РАО, 2002;
20. Сэломон Д. Сжатие данных, изображений и звука. – М.: Техносфера, 2004;
21. Теория информации и кодирование/ Самсонов Б.Б., Плохов Е.М., Филоненков А.И., Кречет Т.В. – Ростов н/Д, 2002;
22. Арапов Д. Пишем упаковщик //Монитор, 1993 – № 1 – С. 16 – 20;
23. Балашов К. Ю. Сжатие информации: анализ методов и подходов. – Минск, 2000;

24. Дискретная математика и математические вопросы кибернетики. Т.1. /Ю.Л. Васильев, Ф. Я. Ветухновский, В. В. Глаголев, Ю. И. Журавлев, В. И. Левенштейн, С. В. Яблонский. Под общей редакцией С. В. Яблонского и О. Б. Лупанова. – М.: Главная редакция физико – математической литературы изд-ва «Наука», 1974;
25. Дмитриев В.И. Прикладная теория информации: Учеб. для студ. вузов по спец. «Автоматизированные системы обработки информации и управления». М.: Высш.шк., 1989;
26. Игнатов В.А. Теория информации и передачи сигналов: Учебник для вузов. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1991;
27. Кричевский Р. Е. Сжатие и поиск информации. – М.: Радио и связь, 1989;
28. Куликовский Л.Ф. и др. Теоретические основы информационных процессов. М.: Высш.шк., 1987;
29. Мастрюков Д. Алгоритмы сжатия информации. Ч. 1. Сжатие по Хаффмену //Монитор, 1993. – № 7 – 8 – С. 14 – 20;
30. Мастрюков Д. Алгоритмы сжатия информации. Ч. 2. Арифметическое кодирование //Монитор, 1994 – № 1 – С. 20 – 23;
31. Мастрюков Д. Алгоритмы сжатия информации. Ч. 3. Алгоритмы группы LZ //Монитор, 1994 – № 2 – С. 10 – 13;
32. Мастрюков Д. Алгоритмы сжатия информации. Ч. 3. Алгоритмы группы LZ //Монитор, 1994 – № 3 – С. 8 – 11;
33. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976;
34. Ризаев И.С. Сборник задач по курсу “Теория информации и кодирование”, Казань, КАИ, 1976;
35. Семенюк В. В. Экономное кодирование дискретной информации. – СПб.: СПб ГИТМО (ТУ), 2001;

36. Смирнов М. А. Использование методов сжатия данных без потерь информации в условиях жестких ограничений на ресурсы устройства – декодера. //www.compression.ru
37. Смирнов М. А. Обзор применения методов безущербного сжатия данных в СУБД //www.compression.ru;
38. Темников Ф.Е. и др. Теоретические основы информационной техники. М.: Энергия, 1979;
39. Фомин А. А. Основы сжатия информации. – СПб.: СПГТУ, 1998.
40. Цымбал В.П. Теория информации и кодирование. Киев: Вища школа, 1977;
41. Экономичное кодирование/ Логинов В.М., Цепков Г.В., Чинаев П.И. – Киев: Издательство «Техніка», 1976.