

Завдання до лабораторних робіт з дисципліни

“Алгоритмізація та програмування”, 2-й семестр

Лабораторна робота № 9 «Структури»

1. Ввести масив записів, який містить прізвища і номери телефонів співробітників. Впорядкувати масив за прізвищами і роздрукувати вхідний і упорядкований масив.
2. Ввести масив записів, який містить прізвища і екзаменаційні оцінки. Записи впорядкувати за прізвищами. Програма обчислює середній бал для кожного прізвища і створює новий масив, що містить прізвища і середні бали.
3. Ввести масив записів, який містить прізвища співробітників і адреси. Записи впорядкувати за адресами. Необхідно ввести новий запис і видалити з масиву ту, яка містить дане прізвище.
4. Ввести масив записів, який містить прізвища авторів книг і назви книг. Створити новий масив записів, який містить додаткову інформацію про рік видання. Рік видання береться з цілочисленого масиву, який впорядкований за прикладом першого масиву.
5. Ввести масив записів, який містить прізвища і відповідні їм адреси. Масив впорядкований за прізвищами. Ввести новий запис і замінити запис, який містить зазначене прізвище, на введену.
6. Ввести масив записів, який містить прізвища співробітників, їх адреси та номери телефонів. Скласти програму, яка переносить ці дані в масив записів, який не містить відомості про номери телефонів.
7. Задані два масиви записів, кожен з яких містить прізвища і адреси співробітників. Складена програма повинна перенести (в будь-якому порядку) записи з двох масивів в третій, причому з двох однакових записів переноситься тільки одна.
8. Задано масив записів, який містить прізвища і адреси. Побудувати новий масив записів, який містить прізвища з початковою літерою С і відповідні їм адреси.
9. Задані два масиви записів, кожен з яких містить прізвища і адреси студентів. Скласти програму, яка друкує однакові записи обох масивів в алфавітному порядку.
10. Задані два масиви записів, кожен з яких містить відомості про прізвища та адреси. Необхідно записи, які є в першому масиві і яких немає в другому масиві, переписати в третій масив в тому ж вигляді.
11. Заданий масив записів, який містить прізвища студентів і їх екзаменаційну оцінку. Скласти програму, яка знаходить і друкує прізвища людей, які отримали задану оцінку.
12. Задані два масиви записів. Перший містить дані - прізвища та адреси, другий - відповідні номери телефонів. Скласти програму, яка об'єднує записи обох масивів і заносить їх в третій масив.
13. Ввести масив записів, які містять відомості-назви і адреси інститутів, програма видаляє записи, заповнені тільки пробілами (пропусками).
14. Заданий масив записів, який містить відомості-прізвища і відповідні їм номери телефонів. Масив впорядкований за прізвищами. Програма повинна вставляти новий запис до відповідного місця масиву в залежності від прізвища, розсуваючи інші елементи масиву.

15. Заданий масив записів, який містить відомості-прізвища і назви груп. Масив впорядкований за прізвищами. Програма видаляє записи, які містять задане прізвище, зрушуючи нижні елементи вгору (заповнюючи ними порожнє місце).
16. Ввести масив записів, який містить відомості-прізвища та номери телефонів. Програма замінює запис, який містить зазначене прізвище, ведення з терміналу. Впорядкувати отриманий масив по прізвищах.
17. Задані два масиви записів, які містять відомості-прізвища та адреси. Масиви впорядковані за прізвищами. Скласти програму, яка переносить в будь-якому порядку записи з двох заданих масивів в третій, причому з двох однакових записів записується тільки одна.
18. Заданий масив записів, який містить відомості-прізвища і назви груп. Масив впорядкувати за прізвищами. Ввести з терміналу назву групи. Переписати в третій масив записи, які мають введену назву групи.
19. Задано два масиви записів, які містять відомості: прізвища та оцінки. Масиви впорядковані за прізвищами. Програма повинна переносити записи першого масиву, які відсутні в іншому масиві, в третій масив в тому ж вигляді.
20. Заданий масив записів, який містять відомості: прізвища та екзаменаційні оцінки. Скласти програму, яка обчислює середній бал всього масиву. Надрукувати вхідний масив і результат обчислення середнього балу.
21. Заданий масив записів, який містить відомості: прізвища студентів і суми їх стипендій. Необхідно видалити з цього масиву всі записи, які мають величину стипендій нижче середнього рівня.
22. Заданий масив записів, який містить відомості: прізвища студентів і рік народження. Необхідно побудувати новий масив записів, в якому будуть тільки записи з роком народження від 1998 і вище.
23. Заданий масив записів, який містить відомості: назва факультету, назва спеціальності. Побудувати новий масив, який доповнений записами, введеними з терміналу і упорядкований по факультетах.
24. Заданий масив записів, який містить відомості: спеціальності, курси, назви груп студентів. Вибрати з цього масиву записи, які відносяться до спеціальності САПР і роздрукувати їх, а також окремо залишок масиву.
25. Заданий масив записів, який містить відомості: спеціальності, курси, групи студентів і середній бал групи. Побудувати новий масив записів, який містить відомості по групах студентів і середній бал, якщо він не менше 4.5. Роздрукувати вхідний і вихідний масиви.
26. Заданий масив записів, який містять відомості: прізвища студентів, їх імена. Надрукувати найбільш поширені чоловічі і жіночі імена.
27. Заданий масиви записів. Один містить відомості: прізвища та імена, інший - прізвища та адреси. Необхідно побудувати новий масив, впорядкований за прізвищами, який містить відомості - прізвища, імена та адреси. Якщо є записи, які не узгоджуються, то повідомити про це.
28. Заданий масив записів, який містить відомості: прізвища студентів і їх екзаменаційні оцінки. Визначити відсоток студентів, які мають середній бал 4.5. Роздрукувати вхідний масив і прізвища студентів з середнім

балом 4.5.

29. У бібліотеці є список книг. Кожен запис цього списку містить прізвища авторів, назва книги, рік видання. Визначити, чи є в даному списку книги, в назві яких зустрічається деяке ключове слово. Якщо є, то видати на друк прізвища авторів, назву і рік видання всіх таких книг. Ключове слово ввести з клавіатури.

30. Написати програму формування відомості про успішність студентів. Кожен запис цієї відомості повинен містити номер групи, прізвище студента, середній бал за останню сесію. Необхідно роздрукувати списки студентів по групах. У кожній групі прізвища студентів розмістити в порядку убутання середнього балу.

Лабораторна робота № 10 «Функції»

1. Дано. дійсні числа a_0, \dots, a_6 . Отримати для $X = 1, 3, 4$ значення

$$p(x+1) - p(x), \text{ де } p(y) = a_6 y^6 + a_5 y^5 + \dots + a_0$$

2. Дано дійсні числа s, t, a_0, \dots, a_{12} . Отримати $p(1) - p(t) + p^2(s-t) - p^3(l)$,

$$\text{де } p(x) = a_{12}x^{12} + a_{11}x^{11} + \dots + a_0$$

3. Дано дійсні числа $a_1, \dots, a_n, b_1, \dots, b_m$. У послідовності a_1, \dots, a_n і послідовності b_1, \dots, b_m всі члени, які йдуть за членом з найбільшим значенням (за першим по порядку, якщо їх декілька), замінити на 0.5.

4. Дано цілі числа $a_1, \dots, a_n, b_1, \dots, b_m, K$. Якщо в послідовності a_1, \dots, a_n немає жодного члена зі значенням K , то перший по порядку член цієї послідовності не менший всіх інших членів, замінити на значення K . За таким же правилом перетворити b_1, \dots, b_m , стосовно до значення 10.

5. Обчислити $z = (s_1 + s_2) / (k_1 + k_2)$, де S_1 і K_1 - сума і кількість позитивних елементів масиву $X(N)$; S_2 і K_2 - сума і кількість позитивних елементів масиву $Y(M)$.

6. Обчислити і запам'ятати суми позитивних елементів кожного рядка матриці $A(10, 20)$, $B(15, 10)$.

7. Обчислити $Z = (X_1 + Y_1) / (X_2 - Y_2)$, де X_1 і X_2 - коріння рівняння

$$2x^2 + x - 4 = 0; Y_1 \text{ і } Y_2 - \text{коріння рівняння } ay^2 + 2y - 1 = 0. \text{ (Всі коріння дійсні).}$$

8. Знайти найбільші елементи і їх порядкові номери масивів $X(N)$ і $Y(M)$.

9. Переписати позитивні елементи $X(10)$ і $Y(8)$ в масив Z поспіль. Запис в масив Z здійснювати в підпрограмі.

10. Знайти найменші елементи і номери рядків і стовпців, в яких вони розташовані, для матриць $A(4, 5)$ і $B(5, 3)$.

11. Вивести на друк елементи цілочисельних матриць $N(3, 6)$ і $M(5, 4)$, кратні трьом.

12. Обчислити і запам'ятати кількість від'ємних елементів кожного стовпця для матриць $A(5, 5)$, $B(4, 5)$.

13. Обчислити суми елементів верхньої трикутної матриці для матриць $A(4, 4)$, $B(5, 5)$.

14. Знайти середні значення, і стандартні відхилення для елементів масивів $X(N)$, $Y(M)$.

15. Обчислити суми і кількості елементів, що знаходяться в інтервалі від a до b для матриць $X(5, 4)$ і $Y(5, 3)$.

16. Перетворити масиви $X(10)$ і $Y(15)$, розташувавши в них поспіль тільки позитивні елементи. Замість інших елементів записати нулі.

17. Обчислити $Z = (e^{s_1} + e^{s_2}) / (k_1 * k_2)$, де S_1 і K_1 - сума і кількість позитивних елементів масиву $X(10)$; S_2 і K_2 - сума і кількість від'ємних елементів масиву $Y(15)$.

18. Обчислити корені квадратного рівняння $x^2 - ax + b = 0$; $cy^2 - dy - f = 0$, використовуючи підпрограму - функцію. (Все коріння дійсні).

19. Обчислити $Z = (X_{\max} - Y_{\min}) / 2$, де X_{\max} - максимальний елемент масиву $X(10)$; Y_{\min} - мінімальний елемент масиву $Y(12)$; X_{\max} і Y_{\min} обчислювати в одній підпрограмі.

20. Обчислити суми позитивних елементів масивів $X(N)$, $Y(M)$, $Z(K)$, використовуючи підпрограму-функцію, де $N < 10$; $M < 12$; $K < 8$.

21. Обчислити $C = (A_{\max} + B_{\min}) / 3$, де A_{\max} і B_{\min} -максимальний і мінімальний елементи відповідно масивів $A(10)$ і $B(12)$.
22. Обчислити суми позитивних елементів кожного рядка для матриць $A(5,6)$ і $B(4,3)$, використовуючи підпрограму-функцію.
23. Обчислити суми елементів головних діагоналей матриць $A(N,N)$, $B(M, M)$, де $N < 10$, $M < 12$, використовуючи підпрограму-функцію.
24. Обчислити суми елементів нижніх трикутних матриць для матриць $A(5,4)$, $B(4,3)$, використовуючи підпрограму-функцію.
25. Скласти процедуру, що дозволяє визначити позицію самого першого входження заданого символу в вихідний рядок. Якщо рядок не містить символу, результатом процедури повинна бути -1.
26. Знайти два рівних елемента в двовимірному динамічному масиві.
27. Виділити слова з рядка і розмістити їх в рядках двовимірного вільного масиву.
28. Знайти максимальний елемент серед мінімальних елементів в рядках двовимірного масиву.
29. Продемонструвати роботу з двовимірним динамічним масивом з використанням функцій введення і виведення.
30. Обчислити суму, різницю і добуток двох верхніх трикутних масивів.

Лабораторна робота № 11 «Символьні дані і рядки. Функції для роботи з рядками»

Мета роботи: вивчити особливості внутрішнього представлення символьних даних і рядків, навчитися виконувати введення/вивід, перегляд і заміну в рядках при рішенні завдань; вивчити синтаксис і семантику стандартних функцій по роботі з рядками, навчитися використовувати в програмах функції для роботи з рядками на мові C.

При виконанні лабораторної роботи для кожного завдання вимагається написати програму на мові C, яка отримує на вході один або декілька рядків символів (залежно від постановки завдання), виконує обробку рядків відповідно до вимог завдання і виводить результат на екран. Введення даних здійснюється з клавіатури з урахуванням вимог до вхідних даних, що містяться в постановці завдання. Обмеженнями на вхідні дані є максимальний розмір строкових даних. Для представлення текстової інформації в мові C використовуються символи (константи), символьні змінні і рядки (строкові константи), для яких в мові C не введено окремого типу на відміну від деяких інших мов програмування.

Для символьних даних в C введений тип `char`. Опис символьних змінних:

```
char список_імен_змінних;
```

Наприклад:

```
const char c='c';
```

```
//символ - займає один байт, його значення не міняється
```

```
char a, b;
```

```
/*символьні змінні, займають по одному байту, значення змінюються*/
```

```
const char *s="Приклад рядка\n";
```

```
//текстова константа
```

Рядок - це послідовність символів, поміщена в подвійні лапки (" ").

Розміщуючи рядок в пам'яті, транслятор автоматично додає в його кінці символ `'\0'` (нульовий символ або нульовий байт, який є ознакою кінця рядка). У записі рядка може бути і один символ: `"A"` (поміщений в подвійні лапки), проте, на відміну від символьної константи `'A'` (використовуються апострофи), довжина рядка `"A"` дорівнює 2 байтам.

У мові C рядок - це пронумерована послідовність символів (масив символів), вона завжди має тип `char[]`. Усі символи рядка нумеруються, починаючи з нуля. Символ кінця рядка також нумерується - йому відповідає найбільший з номерів. Таким чином, рядок прочитується значенням типу "масив символів". Кількість елементів в такому масиві на 1 більше, ніж зображення відповідного рядка, оскільки в кінець рядка доданий нульовий символ `'\0'` (рис. 1).

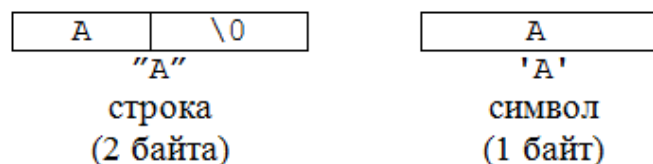


Рис. 1. Представлення рядка і символу

Символьний рядок в програмному коді може розташовуватися на декількох рядках. Для перенесення використовується символ '\ ' з наступним натисненням клавіші введення. Символ '\ ' ігнорується компілятором, і наступний рядок вважається продовженням попереднього.

Присвоїти значення рядку за допомогою оператора присвоювання не можна, оскільки для масивів не визначена операція прямого присвоювання. Помістити рядок в символьний масив можна або при введенні, або за допомогою ініціалізації:

```
char s1[] = "ABCDEF"; //ініціалізація рядка
```

```
char s2[]={ 'A','B','C','D','E','F','\0'}; //ініціалізація рядка
```

Операція обчислення розміру (у байтах) sizeof діє для об'єктів символьного типу і рядків.

Приклад 1.

```
// Визначення розміру рядків
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[]){
    char s1[10]="string1";
    int k=sizeof(s1);
    cout<<s1<<"\t"<<k<<"\n";
    char s2[]="string2";
    k=sizeof(s2);
    cout<<s2<<"\t"<<k<<"\n";
    char s3[]={ 's',' ','t',' ','r',' ','i',' ','n',' ','g',' ','3',' ','\0'};
    k=sizeof(s3);
    cout<<s3<<"\t"<<k<<"\n";
    char *s4="string4";
    k=sizeof(s4);
    cout<<s4<<"\t"<<k<<"\n";
    system("pause");
    return 0;
}
```

Результат виконання програми :

string1 10 - виділено 10 байтів, у тому числі під '\0'

string2 8 - виділено 8 байтів (7 + 1 байт під '\0')

string3 8 - виділено 8 байтів (7 + 1 байт під '\0')

Введення-виведення символівних даних і рядків

1) Введення-виведення поодинокого символу

`getchar()` - функція (без параметрів) використовується для введення поодинокого символу з вхідного потоку. Вона повертає 1 байт інформації (символ) у вигляді значення типу `int`. Це зроблено для розпізнавання ситуації, коли при читанні буде досягнутий кінець файлу.

`putchar(ch)` - функція використовується для виведення поодинокого символу, тобто поміщає в стандартний вихідний потік символ `ch`. Аргументом функції виводу може бути поодинокий символ (включаючи знаки, що представляються управляючі послідовності), змінна або функція, значенням якої є поодинокий символ.

Наприклад:

```
/*Програма прочитує з вхідного потоку один символ, а потім виводить його на екран*/  
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
int _tmain(int argc, _TCHAR* argv[]){  
    char ch;  
    ch=getchar();  
    putchar(ch);  
    system("pause");  
    return 0;  
}
```

Приклад 2. Введіть речення, у кінці якого стоїть точка, і підрахуйте загальну кількість символів, відмінних від пропуску (не рахуючи точки).

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
int _tmain(int argc, _TCHAR* argv[]){  
    char z; //z - символ, що вводиться  
    int k; //k - кількість значущих символів  
    printf("Напишіть речення з точкою у кінці:\n");  
    for (k=0; (z=getchar())!='.');
```

/*вираз `z=getchar()` поміщений у дужки, оскільки операція присвоювання має нижчий ранг ніж операція порівняння*/

```
        if (z!=' ') k++;  
    printf("\nКількість символів=% d", k);  
    system("pause");
```



```
    return 0;
}
```

Результат виконання програми :

Напишіть речення з точкою у кінці:

1 2 3 4 5 6 7 8 9 0.

Кількість символів=10

2) Введення-виведення стандартного текстового (символьного) потоку

gets(s) - функція, яка прочитує рядок s із стандартного потоку до появи символу '\n', сам символ '\n' в рядок не заноситься.

puts(s) - функція, яка записує рядок в стандартний потік, додаючи в кінець рядка символ '\n ', у разі вдалого завершення повертає значення більше або рівне 0 і негативне значення (EOF = - 1) у разі помилки.

Наприклад:

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[]){
    char s[20];
    gets(s);
    puts(s);
    system("pause");
    return 0;
}
```

Результат виконання програми: при введенні рядка "123 456 789" читання даних здійснюється побайтно до символу '\n', тобто в s запишеться рядок "123 456 789\0" (символ '\0' на екран не виводиться, а є ознакою кінця рядка). При виведенні рядка функція puts повертає у кінці рядка додатково один символ '\n', отже, буде виведений рядок "123 456 789\n" (управляючий символ '\n' на екран не виводиться, а здійснює перевід курсора на новий рядок).

Приклад 3. Обчислити довжину введенного рядка.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[]){
    char st[100];
    int i=0;
    puts("Введіть рядок :");
    gets(st);
```

```

while(st[i++] );
printf("Довжина введеного рядка = %i\n", i - 1);
system("pause");
return 0;
}

```

Деякі особливості обробки рядків:

1. Визначення символьного масиву фіксованого розміру:

```
char Array [25];
```

Не можна розмір масиву задавати за допомогою змінної, тобто буде помилкою таке визначення (якщо N не є препроцесорним ідентифікатором):

```
char Array [N];
```

2. Визначення динамічного масиву

```
char *p;
```

```
int N = 25;
```

```
p = (char*) malloc(N);
```

3. Звернення до елемента масиву :

Вирази `Array[0]` і `*Array` - еквівалентні. Також еквівалентні вирази:

```
Array[i], *(Array+i)
```

Для динамічного масиву, адресованого покажчиком `p`, еквівалентні вирази:

```
p[0], *p;
```

```
p[i], *(p+i)
```

Перехід до сусіднього $(i+1)$ -го елемента динамічного масиву:

наступні вирази еквівалентні:

```
p[i+1], p++[i], *(p++ +i)
```

Невірнo: `Array++[i]` - ім'я масиву є константа, за визначенням її не можна змінювати.

4. Варіанти перебору елементів масиву.

```
int i;
```

```
for (i = 0; i<25; i++){ Array[i];}
```

```
for (i = 0; i<25; i++){ *(Array+i);}
```

```
for (i = 0; i<25; i++){ *(p+i);}
```

```
for (i = 0; i<25; i++){ p[i];}
```

Небезпечний варіант (можна забути початок масиву):

```
for (i = 0; i<25; i++) { *p++; }
```

5. Рядки в мові C "розміщуються" в символьних масивах.

Ознакою закінчення рядка служить `'\0'`.

Якщо визначений масив

```
char string[25] = "це рядок";
```

то в масиві string[25] завдовжки 25 байт буде розміщений рядок завдовжки в 11 байт і string[10] == '\0'.

Рядки можуть бути задані в програмі і так:

```
char *ps = "рядок"; /* Довжина рядка 7 байт; ps[6] == '\0'; *(ps+6) == '\0' */
```

```
char str [ ] = "це рядок"; /* Рядок і масив однакової довжини */
```

6. Для введення рядків (що набирають на клавіатурі) зручно використати функцію char* gets(char* s), яка розміщує введену послідовність символів в область пам'яті, що адресується покажчиком s. При цьому ознака кінця введення '\n', що додається у вхідний потік при натисненні клавіші <Enter>, функція gets() замінює ознакою кінця рядка '\0'.

Використання для введення рядків функції scanf("%s") не дозволяє вводити послідовності символів, в яких містяться пропуски або узагальнені пробільні символи. Введення по специфікації %s виконується до найближчого пропуску.

7. Обчислити довжину (len) рядка, що знаходиться в масиві string або адресується покажчиком char *string; можна так:

```
int len;
```

```
for (len=0; *(string+len) !='\0'; len++);
```

Ще зручніше застосовувати функції роботи з рядками із стандартної бібліотеки. Вони описані в заголовному файлі string.h. Наприклад, для визначення довжини рядка використовуйте функцію з прототипом unsigned strlen(char *str). (Повертає довжину рядка без урахування символу '\0', що завершує рядок).

8. Копіювати рядок з масиву string або з рядка, що адресується покажчиком char* string, в новий масив, що адресується покажчиком char* s, можна таким чином:

```
int j;
```

```
char *s; s = (char*) malloc(len);
```

```
for (j=0; j <= len; j++) *(s+j)=*(string+j);
```

де len - довжина початкового рядка.

Варіант, що зміщує покажчик з початку області пам'яті :

```
char *a; a = (char*) malloc(len);
```

```
for (; *a++=*s++);
```

9. Функції з параметрами-рядками:

1) застосування покажчиків

```
int length (const char*) /* const - захист від зміни фактичного параметра */  
{
```

```
    for (int i=0; *(s+i)!='\0'; i++)
```

```
/* Навіть вираз *(s++) не міняє "налаштування" параметра на зовнішній масив */
```

```

    return i; /* Довжина рядка*/
}
2) застосування масиву
int length (char a[]) /*a є const за синтаксисом масиву */ {
    int i=0;
    while (a[i++] != '\0');
    return i - 1; /* Довжина (рядка) */
}

```

Завдання до лабораторної роботи

1. Ввести речення, слова в якому розділені пропусками і у кінці якого стоїть точка. Видалити пропуски, що повторюються, між окремими словами (залишаючи по одному пропуску), виведіть відредагзоване речення на екран.
2. Дано рядок. Зробить символний аналіз тексту. У введеному рядку визначте число: малих українських букв, заголовних українських букв, малих латинських букв, заголовних латинських букв, цифрових символів, інших символів.
3. Напишіть програму, яка перевіряє, чи є введений з клавіатури рядок цілим числом (знак числа не враховувати).
4. Введіть рядок і символ. Визначте частоту появи цього символу в рядку. Частота обчислюється як відношення кількості цих символів в рядку до довжини усього рядка (пропуски враховуються, а символ кінця рядка не враховується).
5. Дано рядок, в якому слова розділені одним пропуском. Замініть перші букви усіх слів на заголовні (якщо слово починається із заголовної букви, залиште без зміни).
6. Дано рядок, в якому слова розділені одним пропуском. Підрахуйте, скільки букв 'а' зустрічається в кожному слові.
7. Дано рядок, в якому слова розділені одним пропуском. Підрахуйте, скільки в кожному слові букв, співпадаючих з його першою буквою.
8. Дано рядок. Перетворіть його так, щоб спочатку були цифрові символи, а потім усі інші. Порядок дотримання символів між собою не змінювати.
9. Задані символи S1, ..., Sn. Відомо, що серед символів S2, S3, .. Sn є декілька знаків питання. Визначити кількість символів пробілу серед Si, ..., Sn, що йдуть перед другим знаком питання.
10. Задано натуральне число n, символи S1, ..., Sn. Отримати перше натуральне i, для якого кожен з символів Si і Si+1 співпадає з буквою "a".
11. Задано натуральне число n, символи S1, ..., Sn. Перетворити послідовність S1, ..., Sn, замінивши в ній кожну точку трьома точками.
12. Задано натуральне число n, символи S1, ..., Sn. Виключити з цієї послідовності усі групи букв виду "abcd".

13. Задано натуральне число n , символи S_1, \dots, S_n серед яких є присутньою двокрапка. Отримати усі символи, які розташовані до першої двокрапки включно.
14. Задано натуральне число n , символи S_1, \dots, S_n Підрахувати найбільшу кількість тих, що йдуть підряд символів пропуску.
15. Задано натуральне число n , символи S_1, \dots, S_n . Визначити кількість входжень в послідовність S_1, \dots, S_n груп букв "abc".
16. Дано натуральне число n , символи S_i, \dots, S_n . Знайти яке-небудь слово, що починається буквою **a** і що закінчується буквою **e** (якщо таких слів немає, то необхідно повідомити про це).
17. Дано натуральне число n , символи S_1, \dots, S_n . Видалити з S_1, \dots, S_n усі слова з непарними порядковими номерами і перевернути усі слова з парними номерами.
18. Дано натуральне число n , символи S_1, \dots, S_n . Видалити з S_1, \dots, S_n усі слова, в яких зустрічається не більше двох різних букв.
19. Дано натуральне число n , символи S_1, \dots, S_n . Видалити з S_1, \dots, S_n усі слова, що закінчуються групою букв, - tur або - ter.
20. Дан рядок. Якщо послідовність S_1, \dots, S_n є паліндромом, тобто $S_1=S_n, S_2=S_{n-1}$, то залишити її без зміни, інакше отримати послідовність $S_1, \dots, S_{n-1}, S_n, S_{n-1}, S_2, S_1$.
21. Дано натуральне число N . Скільки різних цифр зустрічається в його десятковому записі?
22. Перевірити, чи являється в заданому тексті баланс відкриваючих і закриваючих дужок.
23. У введеному реченні вказати слово, в якому доля голосних (A, E, I, O) максимальна.
24. Для пар поруч розташованих символів, що зустрічаються в заданому тексті, вказати, скільки разів зустрічається кожне з таких дволітерних поєднань.
25. Відредагувати речення; видаляючи з нього зайві пропуски, залишаючи тільки по одному пропуску між словами.
26. Для кожного символу заданого тексту вказати скільки разів він зустрічається в тексті.
27. Для кожного слова заданого речення вказати долю приголосних. Визначити слово, в якому доля приголосних максимальна.
28. Відредагувати задане речення, замінюючи багатокрапкову послідовність однією крапкою.
29. У заданому реченні знайти найкоротше і щонайдовше слово.
30. Із заданого тексту вибрати і надрукувати тільки ті символи, які зустрічаються в ньому тільки один раз (у тому порядку, в якому вони зустрічаються в тексті).
31. Знайти щонайдовше симетричне слово заданого речення, наприклад ASSA.
32. У заданому тексті видалити символ ",", і підрахувати число видалених символів.
33. З тексту вибрати числа і записати в масив N . Кількість чисел не більше 10.
34. Видалити з тексту символи "_" і підрахувати довжину сформованого тексту.
35. В тексті замінити символ "_" на символ ",". Якщо в тексті зустрічаються декілька символів "_" підряд, то

замість них поставити одну кому. Визначити довжину речення.

Порядок виконання завдання (загальна схема):

1. Запросити у користувача максимально можливу за умовами довжину LenMax рядка.
2. Створити динамічний символьний масив цього розміру (LenMax).
3. Запросити у користувача початковий рядок і записати його в масив, створений в п.2.
4. Якщо довжина введеного рядка менше LenMax, змінити кількість пам'яті, виділеної під масив в п.2.
5. Виконати дії, необхідні для вирішення завдання.
6. В процесі рішення створити динамічний масив потрібної довжини, що містить результуючий рядок, або декілька масивів (залежно від умов завдання).
7. Надрукувати (вивести на дисплей) результуючий рядок (рядки).
8. Звільнити пам'ять, виділену під усі створені під час роботи динамічні масиви.

Примітка. Якщо введений користувачем початковий рядок містить символи, що не є допустимими, видати повідомлення про помилку і перший неприпустимий символ. Припинити вирішення задачі.

Вказівки до виконання роботи

Кожне завдання необхідно вирішити відповідно до вивчених методів обробки рядкових і символьних даних в мові C. Слід реалізувати кожне завдання відповідно до приведених етапів:

- вивчити словесну постановку завдання, виділивши при цьому усі види даних;
- сформулювати математичну постановку завдання;
- вибрати метод рішення задачі, якщо це необхідно;
- розробити графічну схему алгоритму;
- записати розроблений алгоритм на мові C;
- розробити контрольний тест до програми;
- відлагодити програму;
- представити звіт по роботі.

Відповідно до цілей практикуму в конкретних умовах при виконанні завдань можуть бути введені наступні обмеження:

- не застосовувати індексованих змінних для доступу до елементів масивів-рядків, а використовувати розіменування покажчиків;
- не використовувати стандартні (бібліотечні) функції для роботи з рядками, а явно виконувати усі дії з конкатенації, копіювання, порівняння і ін.

Контрольні питання

1. Чому в C не виконується операція прямого надання значення рядку?
2. Чому символ і рядок, що складається з одного символу, займають різний об'єм пам'яті?
3. Чому у функції `scanf("%s", string);` не вказується звернення до змінної за адресою?
4. Чи допустима операція порівняння над символами? Якщо так, то яким чином визначені стосунки

"більше" і "менше"?

5. Яка з функцій, gets або puts, заносить в потік управляючий символ '\n' і з якою метою?
6. Чи можна виконати присвоювання символьної змінної числового значення? Чому?
7. У чому відмінності результатів виведення символьної змінної із специфікаторами %d і %c?
8. Що буде результатом роботи функції побайтного копіювання рядків, якщо довжина рядка-джерела перевершує допустимий розмір рядка-приймача?
9. Чому при порівнянні рядків важливий регістр символів?
10. Як порівнюються рядки різної довжини?
11. Які можливі наслідки при зверненні до неініціалізованого рядка?
12. Чому функція зміни регістра символів рядка може некоректно працювати з кирилицею?

Лабораторна робота №12 «Робота з рядками в C++»

Мета роботи: вивчити синтаксис і семантику стандартних методів по роботі з рядками, навчитися використовувати в програмах методи для роботи з рядками на мові C++.

Завдання до лабораторної роботи

Варіант 1

Перевірити, чи є вираз, що складається тільки з прописних букв заданого рядка, паліндромом (паліндром – слово або вираз, що читається зліва направо і справа наліво однаково, наприклад, "КАБАК" або "НАЖАЛ КАБАН НА БАКЛАЖАН"). Якщо так, то надрукувати отриманий паліндром. У протилежному випадку вивести рядок, що складається із символів вихідного рядка з видаленими прописними символами.

Можливі символи - цифри; прописні і рядкові латинські букви.

Приклади:

Вихідний рядок:	Результат:
1rK4ABAfGK	KABAK - паліндром
BuRAtino	BRA - не паліндром
AT&T	Неприпустимий символ - '&'

Варіант 2

Маємо рядок, що містить букви і цифри. Перетворіть цей рядок так, щоб спочатку в ньому йшли всі цифри, а потім - усі букви вихідного рядка.

Можливі символи - цифри; прописні і рядкові латинські букви.

Приклади:

Вихідний рядок:	Результат:
ad2e57b6	2576adeb
Tom&Jerry	Неприпустимий символ - '&'

Варіант 3

Маємо рядок, що містить букви і цифри. Перетворіть цей рядок так, щоб спочатку в ньому йшли всі букви, що зустрічаються у вихідному рядку, але в зворотному порядку, а потім - усі цифри вихідного рядка в прямому порядку.

Варіант: спочатку всі букви, потім цифри в зворотному порядку.

Можливі символи - цифри; прописні і рядкові латинські букви.

Приклади:

Вихідний рядок:	Результат:
ad2e57b6	beda2576
KARABAS/BARABAS	Неприпустимий символ - '/'

Варіант 4

Маємо рядок, що містить букви і цифри. Перетворіть цей рядок так, щоб спочатку в ньому йшли всі цифри

вихідного рядка, а потім - усі букви вихідного рядка, але в зворотному порядку.

Можливі символи - цифри; прописні і рядкові латинські букви.

Приклади:

Вихідний рядок:

ad2e57b6

Результат:

2576beda

Strigem.Kozloff.

Неприпустимий символ - '.'

Варіант 5

З клавіатури вводяться:

- речення, слова в якому розділені символом підкреслення ('_');
- маска (шаблон) для вибору з пропозиції потрібних слів (містить букви і символ-заповнювач '*', що заміняє будь-яке сполучення букв, у тому числі порожнє).

Необхідно вибрати з речення всі слова, що відповідають масці (шаблону).

Можливі символи - прописні українські букви; символ-роздільник '_ '.

Приклад:

ВИРАЗ_Є_ПРАВИЛО_ОТРИМАННЯ_ЗНАЧЕННЯ.

Маска:

Підходящі слова:

*Я

ОТРИМАННЯ | ЗНАЧЕННЯ

*Р*З

ВИРАЗ

РА

ВИРАЗ | ПРАВИЛО

Е

ЗНАЧЕННЯ

Варіант 6

Нехай цифрам від 1 до 9 відповідають букви від А (a) до І (i). З клавіатури вводиться рядок. Складіть новий рядок з цифр, що відповідають тільки даним буквам (прописним і рядковим), відсортованим по зростанню.

Можливі символи - прописні і рядкові латинські букви.

Приклад:

Вихідний рядок: SHiFROVkaOtSHPionA

Результат: 1168899 (виділені букви: HiFaHiA)

Варіант 7

З клавіатури вводиться рядок. Виберіть з нього всі букви від А(a) до І(i) (рядкові перетворіть у прописні) і відсортуйте їх за алфавітом .

Можливі символи - прописні і рядкові латинські букви.

Приклад:

Вихідний рядок: SHiFROVkaOtSHPionA

Результат: AAFHHII

Варіант 8

Маємо деяке речення, усі слова якого розділені символами підкреслення (' '). Використовуючи трикутник

Паскаля, зашифруйте вихідне речення за наступним правилом:

- із пропозиції виділяється чергове слово;
- із трикутника Паскаля вибирається рядок з номером, рівним числу букв у слові;
- k-а буква вихідного слова замінюється на набір однакових букв у кількості, зазначений у k-му стовпці обраного рядка трикутника Паскаля.

Трикутник Паскаля:

										1
									1	1
								1	2	1
						1	3	3	1	
			1	4	6	4	1			
	1	5	10	10	5	1				
1	6	15	20	15	6	1				

Можливі символи - прописні українські букви; символ-роздільник '_'.

Приклад:

Вихідний рядок:

"А_РОЗА_УПАЛА_НА_ЛАПУ_АЗОРА"

Зашифрований рядок:

"А_РОООЗЗЗА_УППППААААААЛЛЛЛА_НА_ЛАААПППУ_АЗЗЗЗОООООООРРРРА"

Варіант 12

Перевірити, чи є заданий рядок зашифрований за алгоритмом, що наведено у варіанті 11. Якщо так - видати розшифрований рядок, якщо ні - вивести повідомлення про помилку і вивести перше помилкове слово.

Можливі символи - прописні українські букви; символ-роздільник '_'.

Приклади:

Зашифрований рядок:

НА_ДВВВВОООООООРРРРР_ОССССШШШННННЬ

Результат розшифровки:

НА_ДВОРІ_ОСІНЬ

Зашифрований рядок:

НА_ДВВВВОООООРРРРР_ЗИИИМММА

Результат розшифровки:

Помилка, слово 'ДВВВВОООООРРРРР' не є зашифрованим.

Варіант 13

Виберіть 10 довільних букв українського алфавіту (введіть із клавіатури). Введіть довільне слово. За допомогою ключа довжиною від трьох до восьми символів, що також вводиться користувачем із клавіатури, зробіть шифровку слова в числову комбінацію, як показано в наступному прикладі.

Можливі символи - прописні українські букви (для букв алфавіту і слова), цифри (для ключа).

Приклад:

Нехай заданий набір букв:

В Е Ж М Н О П Р С Т

Поставимо їм у відповідність цифри від 0 до 9:

В Е Ж М Н О П Р С Т

0 1 2 3 4 5 6 7 8 9

Нехай введено яке-небудь слово, наприклад:

М Н О Ж Е С Т В О

Переведемо букви цього слова в цифри, що відповідають цим буквам. У даному випадку:

М Н О Ж Е С Т В О

3 4 5 2 1 8 9 0 5

Запитуємо в користувача цифровий ключ (3 - 8 символів). Нехай користувач увів:

1 2 4 3

Тепер підставляємо ключ під числа, у які переведене наше число, і залишаємо число одиниць результату (десятки відкидаємо). У нашому прикладі:

$$\begin{array}{r} \text{М Н О Ж Е С Т В О} \\ 3 \ 4 \ 5 \ 2 \ 1 \ 8 \ 9 \ 0 \ 5 \\ + \quad 1 \ 2 \ 4 \ 3 \\ + \quad \quad 1 \ 2 \ 4 \ 3 \\ \quad \quad \quad + \ 1 \\ \hline 4 \ 6 \ 9 \ 5 \ 2 \ 0 \ 3 \ 3 \ 6 \end{array}$$

Остаточна відповідь: 4 6 9 5 2 0 3 3 6

Примітка. Якщо довжина ключа не кратна довжині слова, то цифри ключа, що залишилися, відкидаються (як у прикладі).

Варіант 14

Напишіть дешифратор, що перетворить вихідний рядок, зашифрований за принципом, що зазначено у варіанті 13, у слово, використовуючи набір, що вводиться, з десяти можливих символів.

Можливі символи - прописні українські букви (для алфавіту), цифри (для шифру і ключа).

Приклад:

Десять букв: В Е Ж М Н О П Р С Т

Рядок шифру: 4 6 9 5 2 0 3 3 6

Ключ (3 - 8 символів): 1 2 4 3

Розшифрований рядок: М Н О Ж Е С Т В О

Варіант 15

Нехай задане деяке слово. Почати перегляд цього слова зліва направо доти, поки не зустрінуться повторювані букви. Якщо такі букви зустрілися, пропустити їх і продовжити перегляд з кінця слова в зворотному порядку (справа наліво), поки знову не зустрінеться набір повторюваних букв. Якщо такий набір зустрівся, продовжити перегляд з того місця, яке йде за першим набором повторюваних букв і т.д. "Протокол" перегляду рядка вивести на екран; замість послідовності повторюваних букв виводити один символ підкреслення.

Можливі символи - прописні українські букви.

Приклад:

Вихідне слово: НОННИЛЛИОН

"Протокол" перегляду: НО_НОИ_И

Варіант 16

Знайти у вихідному рядку усі входження заданого підрядка і замінити їх на інший рядок з зазначенням номера чергового входження.

Можливі символи - прописні українські букви; символ-роздільник '_'.

Приклад:

Вихідний рядок: ПОЛИЛИ_ЛИЛИЮ

Що замінити: ЛИ

На що замінити: СТО

Результат: ПОСТО1СТО2_СТО3СТО4Ю

Варіант 17

З клавіатури вводиться речення, слова в якому розділені символом підкреслення ('_'). Надрукуйте всі речення, що виходять при перестановці слів вихідного речення.

Можливі символи - прописні українські букви; символ -роздільник '_'.

Приклад:

Вихідне речення: КІШКА_З'ЇЛА_МИШКУ

Перестановки: З'ЇЛА_КІШКА_МИШКУ

МИШКУ_З'ЇЛА_КІШКА

КІШКА_МИШКУ_З'ЇЛА

Варіант 18

1. Зашифруйте речення, що вводиться з клавіатури, у такий спосіб: спочатку вибираються два довільних слова з бази, що знаходиться в тексті програми або вводиться з клавіатури, потім слово із речення, що шифрується, потім знову два слова з бази, після чого - знову слово з речення і т.д. "База" - набір слів, припустимих при виконанні програми, або набір пар слів, як у наведеному нижче прикладі.

2. Напишіть дешифратор.

Можливі символи - прописні українські букви; символ-роздільник '_'.

Приклад:

Речення, що шифрується:

СПРАВА_ЗАКІНЧЕНА_ХАДСОН_РОЗПОВІВ_ВСЕ_СТЕРЕЖІТЬСЯ

База шифру:

З_ДИЧИНОЮ_Я_ВВАЖАЮ_ГОЛОВА_ПІДПРИЄМСТВА_ЗА_ВІДОМОСТЯМИ_ПРО_МУХОБІЙКІ_ФАЗАНЬІХ_КУРОЧОК

Результат шифровки:

З_ДИЧИНОЮ_СПРАВА_Я_ВВАЖАЮ_ЗАКІНЧЕНА_ГОЛОВА_ПІДПРИЄМСТВА_ХАДСОН_ЗА_ВІДОМОСТЯМИ_РОЗПОВІВ_ПРО_МУХОБІЙКІ_ВСЕ_ФАЗАНЬІХ_КУРОЧОК_СТЕРЕЖІТЬСЯ

Варіант 19

З клавіатури вводиться речення, слова в якому розділені символом підкреслення ('_'). Підрахуйте число входжень у речення використовуваних букв, запишіть результат у рядок (парами: буква - цифра) і надрукуйте їх.

Можливими є всі символи, за винятком цифр.

Приклад виконання програми:

Введіть речення: КАРАБАС_БАРАБАС

Результат: А6БЗК1Р2С2

Варіант 20

З клавіатури вводиться задана кількість слів. Знайти усі пари слів, одне з яких закінчується на те ж сполучення букв, на яке починається інше, і вивести результат накладання цих слів один на одного.

Можливі символи - прописні українські букви.

Приклад виконання програми:

Введіть кількість слів: 7

Введіть слова, що розділені пробілами:

ТИНА КОМА ПЛОТ КАМА ОМАР МАЗ ТОК

Результат:

...

КОМАР

ПЛОТИНА

КАМАЗ

Варіант 21

З клавіатури вводиться вихідний рядок.

1. Якщо вихідний рядок не містить цифр (тобто символів з кодами 48 – 57), то він вважається незашифрованим. Вам необхідно замінити всі символи з кодами 33 – 99 на їхні двозначні коди. Символи з іншими кодами виводяться без змін.

Можливі символи - символи з кодами від 33 до 255, за винятком цифр.

Приклад виконання програми:

Введіть вихідний рядок: Hello!

Рядок не зашифрований. Шифр: 72ello33

2. Якщо вихідний рядок містить цифри, то він вважається зашифрованим. Напишіть дешифратор. При неможливості дешифрації надрукуйте повідомлення про помилку і вкажіть місце в рядку, де відбулася помилка.

Можливий діалог із програмою:

Введіть вихідний рядок: 72ello33

Рядок зашифрований. Розшифровка: Hello!

Введіть вихідний рядок: 72el3lo33

Рядок зашифрований. Помилка у вихідному рядку: 72el3

Варіант 22

Дани рядки S1, S2, S3. Знайти кількість символів рядка S3 (без урахування символу '\0'), які є:

- а) одночасно у рядках S1 і S2;
- б) хоча б в одному з рядків S1 або S2;
- в) в жодному з рядків S1 і S2

Варіант 23

З клавіатури вводиться речення, слова в якому розділені символом підкреслення ('_'). Підрахуйте число входжень у речення використовуваних букв, запишіть результат у рядок (парами: буква - цифра) і надрукуйте їх.

Можливими є всі символи, за винятком цифр.

Приклад виконання програми:

Введіть речення: КАРАБАС_БАРАБАС

Результат: А6Б3К1Р2С2

Варіант 24

З клавіатури вводиться задана кількість слів. Знайти усі пари слів, одне з яких закінчується на те ж сполучення букв, на яке починається інше, і вивести результат накладання цих слів один на одного.

Можливі символи - прописні українські букви.

Приклад виконання програми:

Введіть кількість слів: 7

Введіть слова, що розділені пробілами:

ТИНА КОМА ПЛОТ КАМА ОМАР МА3 ТОК

Результат:

...

КОМАР

ПЛОТИНА

КАМАЗ

Варіант 25

З клавіатури вводиться рядок. Виберіть з нього всі букви від Q (q) до Z (z) (рядкові перетворіть у прописні) і відсортуйте їх за алфавітом .

Можливі символи - прописні і рядкові латинські букви.

Приклад:

Вихідний рядок: SHiFROVkaOtSHPionA

Результат: RSSTV

Варіант 26

З рядка S видалити всі слова, які мають у своєму складі хоча б одну цифру.

Варіант 27

Даний рядок, який відображає двійковий запис цілого додатного числа. Вивести рядок, який відображає десятинний запис цього числа.

Варіант 28

Дано рядок – речення. Знайти кількість симетричних слів і вивести рядок з цих слів, які розділені пробілом і розташовані в алфавітному порядку.

Варіант 29

Нехай SET – деяка множина символів. Підрахувати у рядку S кількість різних символів, які приналежать множині SET.

Варіант 30

Перевірити, чи є вираз, що складається тільки з прописних букв заданого рядка, паліндромом (паліндром – слово або вираз, що читається зліва направо і справа наліво однаково, наприклад, "КАБАК" або "НАЖАЛ КАБАН НА БАКЛАЖАН"). Якщо так, то надрукувати отриманий паліндром. У протилежному випадку вивести рядок, що складається із символів вихідного рядка з видаленими прописними символами.

Можливі символи - цифри; прописні і рядкові латинські букви.

Приклади:

Вихідний рядок:

1rK4ABAfGK

BuRAtino

AT&T

Результат:

КАБАК - паліндром

BRA - не паліндром

Неприпустимий символ - '&'

Варіант 31

З клавіатури вводяться:

- речення, слова в якому розділені символом підкреслення ('_');

- маска (шаблон) для вибору з пропозиції потрібних слів (містить букви і символ-заповнювач '*', що заміняє будь-яке сполучення букв, у тому числі порожнє).

Необхідно вибрати з речення всі слова, що відповідають масці (шаблонові).

Можливі символи - прописні українські букви; символ-роздільник '_ '.

Приклад:

ВИРАЗ_Є_ПРАВИЛО_ОТРИМАННЯ_ЗНАЧЕННЯ.

Маска:

Підходящі слова:

*Я ОТРИМАННЯ | ЗНАЧЕННЯ

*Р*З ВИРАЗ

РА ВИРАЗ | ПРАВИЛО

Е ЗНАЧЕННЯ

Лабораторна робота № 13 «Текстові файли в С»

Мета роботи: Вивчити особливості використання текстових файлів, навчитися застосовувати текстові файли в програмуванні.

Визначення та оголошення файлів

Файл як фізичний об'єкт є послідовністю байтів. Фізичний файл характеризується іменем, що його ідентифікує. Максимально можливий розмір файлу залежить від типу файлової системи.

Файл як логічний об'єкт є послідовністю значень певного типу, тобто він складається з однотипних компонентів.

Файли класифікують за типом компонентів і за методом доступу до них. За типом компонентів розрізняють *текстові* та *бінарні* (двійкові) файли, а за методом доступу — файли *послідовного* і *прямого доступу*.

Файл оголошують покажчиком на тип FILE. Синтаксис оголошення *файлової змінної*

```
FILE* <ім'я_показчика>;
```

Структура типу FILE оголошена в заголовному файлі stdio.h.

Відкриття та закриття текстових файлів

Текстовий файл є сукупністю символьних рядків змінної довжини. Кожен рядок завершується *маркером кінця рядка* — спеціальною парою керуючих символів, коди яких: 13 (повернення каретки) та 10 (переведення рядка). Наприкінці файлу записується *маркер кінця файлу* — керуючий символ з кодом 26.

Відповідність між іменем файлу на диску і покажчиками на тип FILE у програмі встановлюється при відкритті файлу. Відкриття файлу здійснюється функцією fopen(). Її прототип оголошено так:

```
FILE *fopen(const char *filename, const char *mode);
```

Функція має такі параметри: filename — рядковий вираз, значенням якого має бути ім'я фізичного файлу; mode — рядковий вираз, значенням якого є режим доступу до файлу. Другий параметр функції fopen() задає режим доступу до файлу та може приймати такі значення: "r" — відкрити файл для читання; "w" — створити новий файл для запису; "a" — відкрити файл для додавання даних у кінець файлу або створити новий файл, якщо він не існує; "r+" — відкрити існуючий файл для читання та запису; "w+" — створити новий файл для запису та читання; "a+" — відкрити файл в режимі читання та запису для додавання нових даних в кінець файлу або створити новий файл, якщо він не існує.

Файли закриваються функцією fclose():

```
fclose(<показчик на файл>);
```

Функція fclose() повертає значення нуль у разі успішного закриття файлу або EOF, якщо виникла помилка.

Зчитування і запис текстових файлів

Зчитування окремих символів із текстового файлу здійснюється функцією fgetc(), яка має прототип:

```
int fgetc(FILE *stream);
```

Зчитування текстового файлу по рядках має такий прототип:

```
char *fgets(char *str, int n, FILE *stream);
```

Тут `str` — буфер, в якому запам'ятовується зчитаний з файлу рядок, заданий як покажчик на тип `char`; `n` — максимальна кількість зчитаних символів; `stream` — покажчик на файл, з якого зчитується рядок.

У мовах C/C++ існують також функції `fputc()`, `fputs()`, що здійснюють запис символів і рядків. Вони мають такі прототипи:

```
int fputc(int c, FILE *stream);
```

```
int fputs(const char *str, FILE *stream);
```

Тут `c` — символ, який записують до файлу; `stream` — покажчик на файл; `str` — покажчик на рядок, який записують до текстового файлу.

Функції зчитування текстових файлів за форматом, який визначає тип значення, що його функція має повернути:

```
int fscanf(FILE *stream, const char *format [,argument ]...);
```

```
int fprintf(FILE *stream, const char *format [,argument ]...);
```

Функції мають параметри: `stream` — покажчик на текстовий файл; `format` — покажчик на рядок формату, який задає правила перетворення символів; `argument` — адреси аргументів, що посилаються на змінні програми.

При зчитуванні з файлу всіх даних потрібно використовувати функцію `feof()`, яка визначає, чи досягнуто кінця файлу:

```
int feof(FILE *stream);
```

Приклад програми

Створити текстовий файл шляхом введення його рядків з клавіатури, в кожному рядку створеного файлу знайти найдовше слово і дописати його в кінець рядка. Вважаємо, що слова відокремлюються одне від одного довільною кількістю пробілів.

Алгоритм задачі

1. Поки не вибрано команду меню `Exit`, відобразити пункти меню на екрані.
2. Вибрати пункт меню.
3. Якщо вибрано пункт меню "Вибрати ім'я файлу", то повторювати такі дії:
 - 3.1. Очистити буфер клавіатури
 - 3.2. Ввести назву файлу для запису
 - 3.3. Ввести назву файлу для читання
4. Якщо вибрано пункт меню "Створення файлу", то повторити такі дії:
 - 4.1. Якщо не задано ім'я файлу, виконати п.3
 - 4.2. Інакше, відкрити файл для запису
 - 4.3. Поки не закінчено введення рядків, повторювати такі дії:
 - 4.3.1. Читати рядок з клавіатури
 - 4.3.2.. Записати рядок до файлу

4.3.3. Записати символ переведення курсору на новий рядок

4.4. Закрити файл

5. Якщо вибрано пункт меню "Читання файлу", то повторити такі дії:

5.1. Вибрати ім'я файлу

5.2. Якщо файл не створено, вивести відповідне повідомлення

5.3. Інакше відкрити файл для читання

5.4. Поки не кінець файлу, виконати такі дії:

5.4.1. Читати рядок з файлу

5.4.2. Вивести рядок на екран

5.5. Закрити файл

6. Якщо вибрано пункт меню "Додати слова до рядків", повторити такі дії:

6.1. Відкрити створений файл для читання

6.2. Відкрити новий файл для запису

6.3. Поки не кінець вхідного файлу, виконати такі дії:

6.3.1. Читати рядок з файлу

6.3.2. Визначити перше слово

6.3.3. Запам'ятати його довжину

6.3.4. Поки не закінчився рядок, повторювати такі дії:

6.3.4.1. Визначити поточне слово та його довжину

6.3.4.2. Якщо довжина поточного слова більша за попередню, запам'ятати довжину поточного слова та саме слово

6.3.5. Приєднати знайдене слово до рядка

6.3.6. Записати рядок до нового файлу

6.4. Закрити файл

Приклад коду

```
/* Пошук найдовшого слова в рядках файлу. Створити текстовий файл шляхом введення його рядків з клавіатури, в кожному рядку створеного файлу знайти найдовше слово і дописати його в кінець рядка. Вважаємо, що слова відокремлюються одне від одного довільною кількістю пробілів. */
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<string.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
FILE* fin,*fout;          //покажчики на вхідний та вихідний файли
```

```
char name[2][20];        //назви вхідного та вихідного файлів
```

```

void Out(FILE* f);

//=====

void ChooseName()
{
puts("enter name of file for write");
fflush(stdin);
gets(name[0]);
puts("enter name of file for read");
fflush(stdin);
gets(name[1]);
}

//===== створення текстового файла =====

void CreateFile()
{
    char answer;                //ознака завершення введення даних
    char s[100];                //рядок, у який вводять дані
    fflush(stdin);
    if(strcmp(name[0],"")==0) ChooseName();
    fout=fopen(name[0],"w");    //відкрити файл для запису
    do
    {
        puts("input string:");    //вивести запит користувачеві
        fflush(stdin);
        gets(s);                //ввести з клавіатури рядок
        fputs(s,fout);          //записати рядок у файл
        fputc('\n',fout);        //записати символ переведення курсору
        puts("continue ? [y/n]");
        answer=getch();          //задати ознаку продовження введення
    }
    while((answer!='n')&&(answer!='N'));
    fclose(fout);              //закрити файл для запису
}

//===== додавання до рядка найдовшого слова =====

void AddWord()
{
    int max;                    //довжина найдовшого слова

```

```

char word[20];                //найдовше слово у рядку
char sa[100], sb[100];       //зчитаний і записаний рядки
char* p;                      //показчик на поточне слово
char* spacer=". ,:";         //роздільники слів у рядках
fin=fopen(name[0],"r");       //відкрити файл для читання
fout=fopen(name[1],"w");      //відкрити файл для запису
while(!feof(fin))            //доки не кінець вхідного файлу
{
    fgets(sa,100,fin);        //читати рядок із файлу
    if (feof(fin)) break;     //якщо кінець файлу, вийти з циклу
    strcpy(sb,sa);            //скопіювати зчитаний з файлу рядок
    p=strtok(sb,spacer);       //перше слово
    max=strlen(p);             //вважати найдовшим
    while (p!=NULL)           //доки у рядку є слова
    {
        if(max<=strlen(p))    //якщо є слово довше за прийняте
        {
            max=strlen(p);     //вважати поточне слово найдовшим
            strcpy(word,p);     //запам'ятати найдовше слово
            if (word[strlen(word)-1]=='\n') //якщо слово
                //останнє в рядку, то його останній символ
                word[strlen(word)-1]=' '; //замінити на пробіл
        }
        p=strtok(NULL,spacer); //визначити наступне слово
    }
    sa[strlen(sa)-1]=' ';      //видалити символ кінця рядка
    strcat(sa,word);           //дописати слово у кінець рядка
    fputs(sa,fout);            //записати рядок у новий файл
    fputc('\n',fout);          //записати символ нового рядка у файл
}                               //кінець циклу читання файлу
fclose(fin);
fclose(fout);
fout=fopen(name[1],"r");
Out(fout);
getch();
fclose(fout);

```

```

}
//===== виведення файлу на екран =====
void Out(FILE* f)
{
    //параметр f - покажчик на файл, що виводиться
    char s[100];                //рядок, зчитаний з файлу
    while(!feof(f))            //доки не кінець файлу
    {
        fgets(s,100,f);        //читати рядок файлу
        if(feof(f)) break;     //якщо кінець файлу, вийти з циклу
        puts(s);               //вивести рядок на екран
    }
    fclose(f);
}
//=====головна програма =====
int main()
{
    char filename[20];
    puts("Search longest word in strings from file");
    system("pause");
    while (1)
    {
        system("cls");
        cout<<"          MENU  \n"<<endl;
        cout<<"1. Choose name of file"<<endl;
        cout<<"2. Create file"<<endl;
        cout<<"3. Read data from file"<<endl;
        cout<<"4. Add word to string"<<endl;;
        cout<<"5. Exit"<<endl;
        cout<<endl;
        int key;                //номер пункту меню
        cout<< "Choosen menu item:";
        cin>>key;
        system("cls");
        switch(key)            // вибір функції реалізації меню
        {
            case 1: {ChooseName();break;}

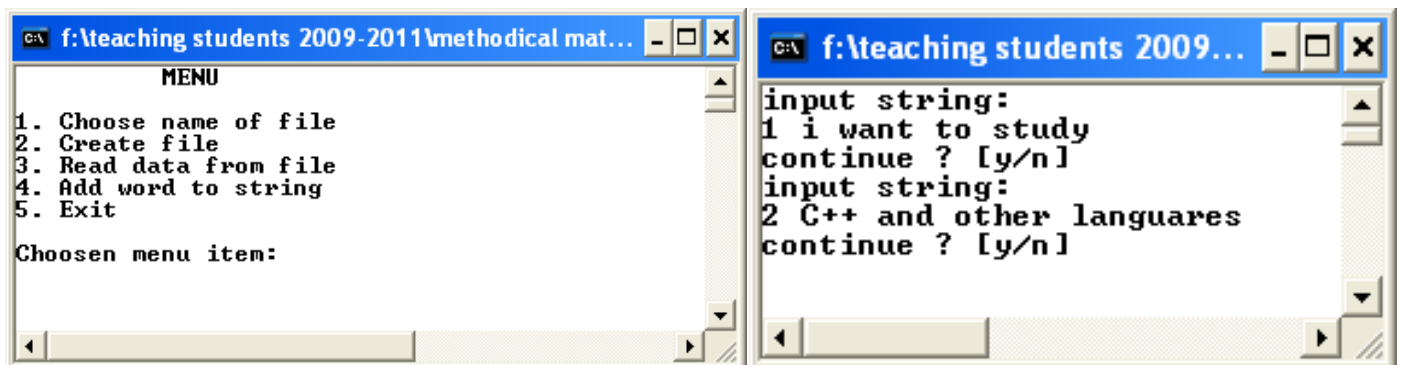
```

```

case 2: {CreateFile();break;}
case 3: {
    puts("choose name of file:");
    ChooseName();
    if ((fin=fopen(name[0],"r"))==NULL) //можливість відкриття файлу
    {
        puts("file not created");
        strcpy(filename,name[1]);
    }
    else
        if ((fin=fopen(name[1],"r"))==NULL)
        {
            puts("file not created");
            strcpy(filename,name[0]);
        }
    if ((fin=fopen(filename,"r"))==NULL) puts("return menu");
    else
    {
        cout<<"text from file '" <<filename<<"':"<<endl;
        Out(fin);
    }
    getch();break;
}

case 4: {AddWord(); break;}
case 5: {exit (0); break;}
default: {cout<<"Try another time...";getch();break;}
}
}
}

```



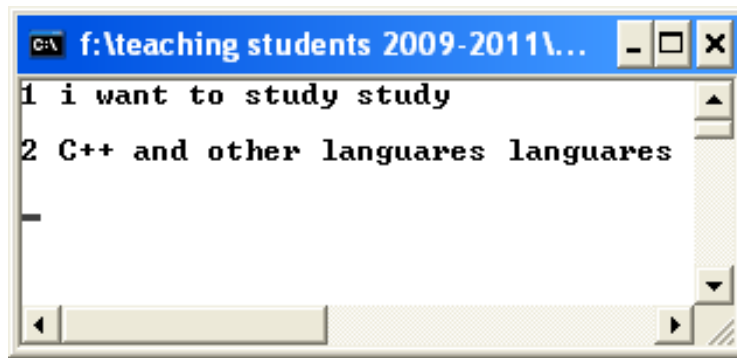


Рис. 2. Результати роботи програми обробки текстових файлів

Варіанти завдань

1. Створити текстовий файл, кожний рядок якого містить різні слова. Знайти найкоротше слово в кожному рядку файлу, переписати їх в новий текстовий файл, записати останнім рядком файлу кількість слів у файлі.
2. Створити текстовий файл, рядок якого містить цифри і символи. В кожному рядку визначити найбільшу послідовність цифр, що йдуть поспіль. Значення довжин цифрових послідовностей записати першими символами рядків вхідного файлу.
3. Створити текстовий файл. Визначити в кожному парному рядку слово найбільшої довжини і записати це слово в кожному непарному рядку текстового файлу.
4. Створити текстовий файл, задавши з клавіатури назву фірми, марку автомобіля, його ціну. Визначити максимальну ціну автомобіля, що випускається кожною фірмою і дописати це значення в кінець кожного рядка файлу.
5. Ввести з клавіатури рядки і записати їх у текстовий файл. У кожному непарному рядку визначити слово, що має найбільшу кількість головних. Дописати знайдені слова в кожний рядок файлу.
6. Ввести з клавіатури декілька рядків тексту та записати їх у текстовий файл. Визначити в кожному рядку кількість слів і розділових символів. Дописати значення кількості слів на початок кожного рядка, значення кількості розділових символів в кінець рядка файлу.
7. Ввести рядки з клавіатури і записати їх у текстовий файл. Визначити кількість рядків файлу, максимальну довжину рядка, кількість порожніх рядків, рядок, що починається та завершується заданими користувачем символами. Отримані значення записати у новий текстовий файл.
8. Створити текстовий файл, перший рядок якого містить значення n та m , які визначають кількість рядків та стовпців матриці. Наступні n рядків містять по m чисел, що є елементами матриці. Визначити максимальне значення в кожному рядку матриці та дописати їх в кожний рядок файлу, починаючи з другого.
9. Створити текстовий файл, рядки якого містять дані про назву фірми, назву товару, ціну в доларах. Перерахувати ціни в \$ у відповідності з курсом в гривні та дописати в кожний рядок файлу отримані значення.
10. Створити текстовий файл. Визначити кількість рядків, які починаються з заданого користувачем символу, починаються та закінчуються одним й тим самим символом, складаються з однакових символів, є пустими рядками. Отримані значення записати в окремий файл з відповідними коментарями.

11. Створити текстовий файл F. Увести з клавіатури рядок символів S. Отримати всі рядки файлу F, що містять у собі рядок S і записати їх до нового файлу G. Останнім рядком файлу G записати кількість знайдених у файлі F рядків.
12. Створити текстовий файл, який містить додатні, від'ємні, нульові числа та довільні символи. Визначити кількість додатних, від'ємних, нульових чисел та слів у кожному рядку файлу. Записати отримані значення з відповідними коментарями в інший текстовий файл.
13. Створити текстовий файл, який би містить марку автомобіля, його вартість, кілометраж. Упорядкувати рядки за алфавітом марки автомобіля, в кожному рядку файлу додати середній кілометраж по всім маркам у файлі. Останній рядок нового файлу має містити кількість різних перерахованих у файлі автомобілів.
14. Створити текстовий файл, який містить числа і символи. Визначити в кожному рядку файлу середнє арифметичне та середнє геометричне чисел рядка. Переписати в новий текстовий файл числа з попереднього файлу та їх середні значення.
15. Створити два текстових файлів. Рядки першого файлу містять назву продукту та його ціну. Рядки другого файлу містять значення кількості кожного продукту. Створити третій текстовий файл, кожен рядок якого має містити назву продукту, його кількість та ціну. У третьому файлі рядки повинні містити назви товарів, що не повторюються.
16. Створити текстовий файл, рядки якого містять числа, що задають координати вершин трикутника. Визначити вид трикутника за значеннями його сторін. Дописати в кожен рядок вхідного файлу слово, що визначає вид трикутника (прямокутний, рівнобедрений тощо).
17. Створити текстовий файл, рядки якого містять по 3 числа, що задають коефіцієнти рівняння прямої в декартових координатах ($Ax+By+C=0$) на площині. Визначити лінії, що є паралельними, перпендикулярними та ті, що перетинаються. Створити новий текстовий файл, в рядки якого записати по 6 чисел – коефіцієнтів прямих та слово, що визначає їх взаємне розташування.
18. Створити текст, вводячи дані з клавіатури. У кожному рядку тексту знайти слово максимальної довжини і записати знайдені слова та їх довжини у новий текстовий файл.
19. Створити програму, що перевіряє правильність розстановки фігурних дужок у текстовому файлі, що є програмою на C.
20. Створити текстовий файл F. Переписати рядки файлу F до файлу H. При цьому вставити на початку рядка порядковий номер, в кінці рядка – його довжину. Порядок рядків зберігається.
21. Створити текстовий файл. Вилучити з файлу всі слова, довжина яких менше заданої користувачем, та пусті рядки. Записати в останній рядок текстового файлу, кількість вилучених слів.
22. Створити два текстових файлів, перший рядок яких містить значення n та m, які визначають кількість рядків та стовпців матриці. Наступні n рядків містять по m чисел, що є елементами матриці. Вивести на екран матриці. Визначити добуток двох матриць, результати множення матриць вивести на екран та у новий текстовий файл.

23. Створити текстовий файл. Отримати його копію. У вхідному файлі замінити усі входження одного слова на інше. Слова задаються користувачем з клавіатури.
24. Створити два тестових файлів. Визначити та вивести рядки, які співпадають в обох файлах, та кількість рядків, що відрізняються.
25. Створити текстовий файл F. Записати до файлу G рядки файлу F в зворотному порядку, виключивши заданий користувачем символ.
26. Ввести з клавіатури декілька рядків тексту та записати їх у текстовий файл. Визначити в кожному рядку кількість слів і розділових символів. Дописати значення кількості слів на початок кожного рядка, значення кількості розділових символів в кінець рядка файлу.
27. Ввести рядки з клавіатури і записати їх у текстовий файл. Визначити кількість рядків файлу, максимальну довжину рядка, кількість порожніх рядків, рядок, що починається та завершується заданими користувачем символами. Отримані значення записати у новий текстовий файл.
28. Створити текстовий файл, який містить додатні, від'ємні, нульові числа та довільні символи. Визначити кількість додатних, від'ємних, нульових чисел та слів у кожному рядку файлу. Записати отримані значення з відповідними коментарями в інший текстовий файл.
29. Створити текстовий файл, який містить числа і символи. Визначити в кожному рядку файлу середнє арифметичне та середнє геометричне чисел рядка. Переписати в новий текстовий файл числа з попереднього файлу та їх середні значення.
30. Створити текст, вводячи дані з клавіатури. У кожному рядку тексту знайти слово максимальної довжини і записати знайдені слова та їх довжини у новий текстовий файл.

Контрольні питання

1. Чим відрізняються бінарні файли від текстових?
2. Чим відрізняється послідовний доступ до компонентів файлу від прямого доступу?
3. У чому полягає відмінність між по символічній обробкою текстових файлів від обробки по рядках?
4. Як зв'язати логічний файл із фізичним?
5. Які способи зчитування та запису текстових файлів надає мова C/C++?
6. Які поля містить структура FILE?

Лабораторна робота № 14 «Структури та бінарні файли»

Мета роботи: Вивчити особливості використання структур, об'єднань та файлів, навчитися застосовувати структури, об'єднання, файли в програмуванні.

Визначення та оголошення структур і об'єднань

Структура — це складений тип даних, що являє собою об'єднання фіксованої кількості змінних одного або декількох типів. Змінну типу структури називають *екземпляром* структури. Синтаксис оголошення типу структури такий:

```
struct <ім'я типу структури> {  
    <тип> <ім'я поля1>;  
    ...  
    <тип> <ім'я поляN>;  
};
```

Екземпляри структури можуть подаватися як змінні інших типів — змінні типу структури, покажчики на структури, масиви структур:

```
<ім'я типу структури> <ім'я змінної>;  
<ім'я типу структури>* <ім'я покажчика>;
```

Звернення до полів структур здійснюється через *складене ім'я*, що має такий синтаксис:

```
<ім'я змінної типу структури>.<ім'я поля>
```

У разі оголошення покажчика на структуру доступ до полів змінної типу структури здійснюється за допомогою *операції стрілка* -> через складене ім'я:

```
<ім'я покажчика на тип структури> -> <ім'я поля>
```

В мовах C/C++ є різновид структури, що зберігає елементи в адресному просторі, що розділяється. Такі структури називаються *об'єднанням*. Особливостями об'єднання є те, що його елементи розміщуються за одною адресою, тобто перекривають один одного. Очевидно, що розмір пам'яті, що виділяє компілятор, визначається розміром найбільшого елемента об'єднання.

Оголошення об'єднання має такий синтаксис:

```
union [<ім'я типу>]{  
    <список_1 полів>;  
    ...  
    <список_N полів>;  
    }[<ідентифікатор>;
```

У разі, коли об'єднання є складовою частиною іншої структури, поле ознаки, яке визначатиме, котрий з елементів об'єднання активізується, входить до складу цієї структури.

Зчитування та запис бінарних файлів

Під час відкриття бінарних файлів викликають функцію `fopen()`, проте вказують режим бінарної обробки: "rb" — тільки зчитування, "wb" — тільки запис, "ab" — доповнення в кінець, "r+b" і "w+b" позначають режим запису та зчитування.

Приклад відкриття бінарного файлу "MyFile.dat" з покажчиком `FILE* fp`:

```
fp=fopen("MyFile.dat", "r+b");
```

Прототипи функцій зчитування та запису в бінарні файли є такими:

```
size_t fread(void *buffer, size_t size, size_t count, FILE *stream);
```

```
size_t fwrite(const void *buffer, size_t size, size_t count, FILE *stream);
```

Тут `buffer` — адреса змінної, з якої байти копіюються на диск під час виклику `fwrite()`, або адреса змінної, в яку байти зчитуються під час виклику `fread()`; `size` — кількість байтів, що виділені під змінну; `count` — кількість елементів розміру `size`, що записуються на диск або зчитуються з диска; `stream` — покажчик на файловий потік, відкритий у бінарному режимі.

Прямий доступ до компонентів бінарних файлів

Наведені далі функції мов C/C++ реалізують прямий метод доступу.

Функція `fgetpos()` зберігає позицію файлового покажчика у змінній, що адресується другим параметром. Значення позиції файлового покажчика дорівнює зміщенню в байтах від початку файлу до позиції, з якої має розпочатися наступна операція введення-виведення. Прототип функції такий:

```
int fgetpos(FILE *stream, fpos_t *pos);
```

Аргументи: `stream` — покажчик на файловий потік; `pos` — покажчик на поточну позицію у файлі; тип `fpos_t` є перейменуванням (аліасом) типу `long`. У разі успішного виклику функції вона повертає значення нуль, інакше — -1.

Функція `fsetpos()` відновлює позицію файлового покажчика, встановлюючи його рівним значенню, що його отримала функція `fgetpos()`. Параметри функції `fsetpos()` аналогічні розглянутим вище, значення `pos` визначається попереднім викликом `fgetpos()`. Функція повертає значення нуль, якщо її робота успішна, та ненульове значення в разі помилки. Прототип функції `fsetpos()` оголошений так:

```
int fsetpos(FILE *stream, const fpos_t *pos);
```

Отримати поточну позицію файлового покажчика, яка згодом використовуватиметься для переміщення файлового покажчика на потрібний байт, можна за допомогою функції `ftell()` з прототипом:

```
long ftell( FILE *stream);
```

Функція `ftell()` повертає позицію файлового покажчика. Значення, що його повертає функція `ftell()`, слід передати у функцію `fseek()`, яка переміщує файловий покажчик на певну кількість байтів вперед по файлу (від його початку) або назад (від його кінця). Функція має такий прототип:

```
int fseek( FILE *stream, long offset, int origin);
```

Тут `stream` — покажчик на тип `FILE`; `offset` — кількість байтів, на які слід перемістити файловий покажчик

даних у напрямку, що вказаний третім параметром; `origin` — напрям переміщення файлового покажчика, задається константами `SEEK_SET` для переміщення файлового покажчика в напрямку від початку файлу, `SEEK_END` для переміщення файлового покажчика в напрямку від кінця файлу, `SEEK_CUR` для переміщення файлового покажчика відносно його поточної позиції.

Приклад програми

Створити масив структур. Кожна структура складається з таких елементів: прізвище викладача, дисципліна, прізвище студента. Для викладача задається посада, зарплата, для студента - середній бал, стипендія. Створений масив структур записати до файлу. Реалізувати запити: визначити викладачів і дисципліни з найвищим середнім балом; перерахувати стипендію таким чином: вилучити стипендію у студентів, середній бал яких нижче за трійку, і розподілити їх стипендію серед інших студентів; визначити стипендіальний фонд (сумарний розмір стипендії) і порівняти його з фондом зарплати викладачів.

Усі дані для запитів беруться з файлу.

Алгоритм задачі

1. Поки не вибрано команду меню `Exit`, відобразити пункти меню на екрані.
2. Вибрати пункт меню.
3. Якщо вибрано пункт меню "Введення початкових даних у масив структур", то повторювати такі дії:
 - 3.1. Ввести назву дисципліни, прізвище викладача, його посаду, зарплату;
 - 3.2. Для кожного викладача повторювати введення з клавіатури прізвища, середнього балу та стипендії студентів;
 - 3.3. Якщо натиснута клавіша `<n>` для припинення введення, то кінець введення даних.
4. Якщо вибрано пункт меню "Читання даних з файлу", то повторити такі дії:
 - 4.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
 - 4.2. Інакше, поки не кінець файлу, зчитувати дані з файлу до структури у оперативну пам'ять;
 - 4.3. Вивести дані, згруповані за дисциплінами;
 - 4.4. Читати наступний запис з файлу до структури;
 - 4.5. Якщо кінець файлу, то закрити його.
5. Якщо вибрано пункт меню "Виведення відомостей про викладачів", то повторювати такі дії:
 - 5.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
 - 5.2. Інакше зчитати дані з файлу до структури записів;
 - 5.3. Вивести дані на екран;
 - 5.4. Якщо не кінець файлу, то зчитати наступний запис з файлу до структури;
 - 5.5. Інакше закрити файл.
6. Якщо вибрано пункт меню "Виведення відомостей про студентів", то повторювати такі дії:
 - 6.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
 - 6.2. Інакше зчитати дані з файлу до структури;

- 6.3. Вивести відомості про студентів;
- 6.4. Якщо не кінець файлу, то зчитати наступний запис з файлу до структури;
- 6.5. Інакше закрити файл.
7. Якщо вибрано пункт меню "Пошук дисципліни з найвищим середнім балом", то повторювати такі дії:
 - 7.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
 - 7.2. Інакше зчитати з файлу елемент масиву структур;
 - 7.3. Накопичення суми балів студентів з окремої дисципліни;
 - 7.4. Знайти середній бал успішності з конкретної дисципліни;
 - 7.5. Якщо не кінець файлу, то зчитати наступний елемент масиву структур;
 - 7.6. Інакше закрити файл;
 - 7.7. Вивести середні бали з усіх дисциплін;
 - 7.8. Знайти максимальний середній бал серед дисциплін;
 - 7.9. Вивести остаточні результати.
8. Якщо вибрано пункт меню "Перерозподіл стипендій", то повторювати дії:
 - 8.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
 - 8.2. Інакше, поки не кінець файлу, зчитати дані з файлу до структури;
 - 8.3. Формувати масив даних про студентів;
 - 8.4. Якщо кінець файлу, то закрити файл.
 - 8.5. Цикл перебирання масиву студентів:
 - 8.5.1. Накопичення суми балів з різних дисциплін для конкретного студента;
 - 8.5.2. Розрахувати кількість дисциплін, які вивчає конкретний студент;
 - 8.5.3. Розрахувати середній бал з усіх предметів для конкретного студента;
 - 8.5.4. Сортування прізвищ студентів за алфавітом;
 - 8.5.5. Вивести дані про загальний середній бал і введену стипендію;
 - 8.5.6. Якщо оброблено дані про останнього студента, то кінець циклу, інакше перейти до наступного студента.
 - 8.6. Обробити дані про студентів з середнім балом меншим за трійку:
 - 8.6.1. Розрахувати суму стипендій студентів, що мають середній бал, менший трьох;
 - 8.6.2. Розрахувати кількість студентів, яким треба перерахувати стипендію.
 - 8.6.3. Цикл перебирання масиву студентів:
 - 8.6.3.1. Якщо середній бал від трійки до п'ятірки, то до початкової стипендії додається стипендія, яка знімається з студентів із середнім балом, меншим за трійку;
 - 8.6.3.2. Вивести результати перерозподілу стипендій;
 - 8.6.3.3. Якщо дані по всіх студентах оброблено, то кінець циклу.
 9. Якщо вибрано пункт меню "Порівняння стипендіального фонду і фонду зарплат викладачів", то

повторювати такі дії:

- 9.1. Якщо файл не існує, то вивести попереджувальне повідомлення;
- 9.2. Інакше, доки не кінець файлу, зчитати дані з файлу до структури;
- 9.3. Накопичення суми зарплатні викладачів;
- 9.4. Якщо прочитано останній запис з файлу, то закрити файл.
- 9.5. Цикл переглядання масиву студентів:
 - 9.5.1. Накопичення суми стипендій;
 - 9.5.2. Якщо вибрано останнього студента, то кінець циклу.
- 9.6. Вивести результати порівняння.
10. Якщо вибрано пункт меню "Вихід", то завершити програму.
11. Якщо вибрано неіснуючий пункт меню, то вивести відповідне повідомлення.

Приклад коду (без використання об'єднання):

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
#include<iomanip>
using namespace std;
struct teacher          //структура викладач
{
    char fio[10];        //прізвище
    char posada[20];     //посада
    float salary;        //зарплатня
};
struct stud             //структура студента
{
    char fio[10];        //прізвище
    float bal;           //середній бал
    float stip;          //стипендія
};
struct subject          //структура дисципліни
{
    char name[10];       //назва дисципліни
    teacher prepod;      //викладач, що читає предмет
    int stnum;           //кількість студентів, що вивчають дану дисципліну
```



```

stud student[10] ;    //масив студентів, що вивчають предмет
};

int i,j;              //лічильники викладачів і студентів
FILE *fp;             //показчик на файл
//-----введення даних-----

void input()
{
    subject predmet[10];    //масив дисциплін
    char answer_prep, answer_stud;
    if (fp!=NULL)           //якщо файл уже існує
        remove("fp.dat");    //знищення файлу
    else
    {
        fp=fopen("fp.dat","wb");//відкриття файлу для запису
        i=0;                //початкове значення лічильника
        do
        {
            cout<<"subject name"<<endl;
            cin>>predmet[i].name;
            cout<<"teacher's surname"<<endl;
            cin>>predmet[i].prepod.fio;
            cout<<"teacher's position"<<endl;
            cin>>predmet[i].prepod.posada;
            cout<<"teacher's salary"<<endl;
            cin>>predmet[i].prepod.salary;
            j=0;
            do
            {
                cout<<"student's surname"<<endl;
                cin>>predmet[i].student[j].fio;
                cout<<"student's ball on subject"<<endl;
                cin>>predmet[i].student[j].bal;
                cout<<"student's scholarship"<<endl;
                cin>>predmet[i].student[j].stip;
                j++;
                cout<<"any students else? y/n"<<endl;
            }
        }
    }
}

```

```

    answer_stud=getch();
}
while(answer_stud!='n');
predmet[i].stnum=j;                //кількість студентів по дисципліні
cout<<"contunue input? y/n"<<endl;
answer_prep=getch();
fwrite(&predmet[i],sizeof(subject),1,fp); //запис предмету до файлу
i++;                                //перехід до нового предмету
}
while (answer_prep!='n');
fclose(fp);                        //закриття файлу
}
getch();                          //чекати натискання клавіші
}
//-----виведення даних по викладачах-----
void list_prepod()
{
    subject predmet;
    if (!(fp=fopen("fp.dat","rb"))) cerr<<"File do not exist!";
    else
    {
        cout<<"---subject---teacher-----position-----salary \n";
        fread(&predmet,sizeof(subject),1,fp); //читання даних з 1 предмету
        while(!feof(fp))                    //доки не кінець файлу
        {
            cout.setf(ios::left); //виведення даних по викладачах
            cout<<"      "<<setw(10)<<predmet.name<<setw(12)<<predmet.prepod.fio<<
            setw(14)<<predmet.prepod.posada<<predmet.prepod.salary<<endl;
            fread(&predmet,sizeof(subject),1,fp); //читання даних з 1 предмету
        }
        fclose(fp);                        //закриття файлу
    }
    getch();                              //чекати натискання клавіші
}
//-----виведення даних по студентах-----
void list_student()

```

```

{
    subject predmet;                                //дисципліна
    if (!(fp=fopen("fp.dat","rb")))//перевірка існування файлу при відкритті
        cerr<<"File do not exist!";
    else
    {
        cout<<"---teacher----student-----ball---scholarship \n";
        fread(&predmet,sizeof(subject),1,fp);    //читання даних з 1 предмету
        while(!feof(fp))
        {
            cout<<"    "<<predmet.prepod.fio<<endl;
            for(j=0;j<predmet.stnum;j++)          //перегляд студентів по даній
            {                                       // дисципліні
                cout.setf(ios::left);              //виведення даних по студентах
                cout<<"                "<<
                    setw(12)<<predmet.student[j].fio<<setw(10)<<
                    predmet.student[j].bal<<predmet.student[j].stip<<endl;
            }
            fread(&predmet,sizeof(subject),1,fp); //читання даних з 1 предмету
        }
        fclose(fp);                                //закриття файлу
    }
    getch();                                         //чекати натискання клавіші
}
//-----читання даних з файлу-----
void read_from_file()
{
    subject predmet;
    if (!(fp=fopen("fp.dat","rb")) cerr<<"File do not exist!";
    else
    {
        cout<<setw(20)<<"It is file:"<<endl;
        fread(&predmet,sizeof(subject),1,fp);    //читання даних з 1 предмету
        while(!feof(fp))
        {                                           //виведення даних з файлу
            cout<<"subject-----teacher----position-----salary"<<endl;

```

[illegible]

```

        i++;                                //перехід до наступного предмету
        fread(&predmet[i],sizeof(subject),1,fp); //читання даних з 1
                                                //предмету
    }
    int kol_prep=i;                          //кількість викладачів
    cout<<" Subject\'s average ball"<<endl;
    for (i=0;i<kol_prep;i++)
    {
        cout.setf(ios::left);
        cout<<setw(10)<<predmet[i].name<<setw(5)<<sr[i]<<endl;
    }
    cout<<"subject with the highest ball: "<<endl;
    float max=sr[0];                        //максимальний бал
    int nomer_predmet=0;                    //номер дисципліни(викладача) у масиві
                                                //дисциплін
    for(i=1;i<kol_prep;i++)                //перегляд дисциплін
    if (max<sr[i])
    {
        max=sr[i];                        //визначити максимальний бал й номер дисципліни
        nomer_predmet=i;
    }
    cout<<"max ball="<<max<<" in subject  "<<predmet[nomer_predmet].name
    <<" teacher "<<predmet[nomer_predmet].prepod.fio<<endl;
    fclose(fp);                            //закриття файлу
}
getch();                                  //чекати натискання клавіші
}
//----- перерозподіл стипендій-----
void stipendia()
{ float sum=0;    //сума стипендій студентів с середнім балом менше 3
  int k=0;        //кількість записів про студентів у вхідному масиві
  int l=0;        //лічильник всіх студентів
  subject predmet;    //предмет
  stud s,tmp[10],stud_tmp[10]; //проміжні масиви студентів
  //копіювання даних по студентам у тимчасовий масив
  // без дисциплін та даних по викладачам

```

```

if (!(fp=fopen("fp.dat","rb"))) cerr<<"File do not exist!";
else
{ fread(&predmet,sizeof(subject),1,fp);
  while(!feof(fp))
  {
    for(j=0;j<predmet.stnum;j++) //перегляд студентів по даній дисципліні
    {
      //копіювання масиву студентів
      tmp[k]=predmet.student[j];
      k++;
    }
    fread(&predmet,sizeof(subject),1,fp);
  }
  fclose(fp); //закриття файлу
  //середній бал студентів з урахуванням вивчення різних предметів
  int kol[10]; //лічильник предметів, що вивчає один студент
  for (i=0;i<k;i++) //перегляд всього масиву студентів
  {
    stud_tmp[l].bal=0; //сума балів одного студента по різних предметам
    kol[l]=0; //кількість предметів для одного студента
    int f=0; //прапорець обробки даних по студенту
    for (j=0;j<i;j++)//перегляд студентів спочатку масиву до даного запису
    if ((strcmp(tmp[i].fio,tmp[j].fio)==0))
      f=1; //студент вже зустрічався в масиві
    if (f==0) //студент іще не зустрічався в масиві
    {
      for (j=i;j<k;j++) // перегляд масиву від даного запису до кінця
        if (strcmp(tmp[i].fio,tmp[j].fio)==0) //чи є той самий студент
          //по іншому предмету
        {
          stud_tmp[l].bal+=tmp[j].bal; //накопичення балів
          //по одному студенту
          strcpy(stud_tmp[l].fio, tmp[j].fio); //запам'ятати студента
          stud_tmp[l].stip=tmp[j].stip; //запам'ятати стипендію
          kol[l]++; //перехід до наступної дисципліни
        }
      //середній бал студента по всім предметам
    }
  }
}

```

```

        stud_tmp[l].bal=float(stud_tmp[l].bal)/kol[l];
        l++;                      //перехід до наступного студента
    }
}

int all_stud=1;                  //загальна кількість студентів
//сортування студентів за алфавітом
for (i=0;i<all_stud;i++)
    for(j=i+1;j<all_stud;j++)
        if (strcmp(stud_tmp[i].fio,stud_tmp[j].fio)>0)
        {
            s=stud_tmp[i];
            stud_tmp[i]=stud_tmp[j];
            stud_tmp[j]=s;
        }

//виведення даних про загальний середній бал та початкову стипендію
cout<<"Array of students after sort"<<endl;
cout<<"--student-----average ball-----scholarship"<<endl;
for (i=0;i<all_stud;i++)
{
    cout.setf(ios::left);
    cout<<setw(20)<<stud_tmp[i].fio<<setw(15)<<stud_tmp[i].bal<<
    stud_tmp[i].stip<<endl;
}
getch();

//обробка студентів з середнім балом менше 3
j=0;                             //їх кількість
for (i=0;i<all_stud;i++)
    if ((stud_tmp[i].bal>=0)&&(stud_tmp[i].bal<3))
    {
        sum+=stud_tmp[i].stip;
        stud_tmp[i].stip=0;        //обнулення їх стипендії
        j++;
    }

for (i=0;i<all_stud;i++)          //розподіл між іншими
    if ((stud_tmp[i].bal>=3)&&(stud_tmp[i].bal<=5))
        stud_tmp[i].stip=stud_tmp[i].stip+sum/(all_stud-j);

```

```

//виведення нарахованої стипендії
cout<<endl<<"Counted scholarship"<<endl;
cout<<"--FIO-----average ball-----scholarship"<<endl;
for(i=0;i<all_stud;i++)
{
    cout.setf(ios::left);
    cout<<setw(15)<<stud_tmp[i].fio<<setw(15)<<stud_tmp[i].bal;
    if ((stud_tmp[i].bal>=3)&&(stud_tmp[i].bal<=5))
        cout<<stud_tmp[i].stip<<endl;
    else
        cout<<"has no scholarship"<<endl;    //стипендія не нарахована
}
}
getch();    //чекати натискання клавіші
}
//-----стипендіальний фонд-----
void fond()
{
    float fond_prep=0;    //фонд викладачів
    float fond_stip=0;    //стипендіальний фонд
    subject predmet;    //предмет
    stud tmp[10];    //тимчасовий масив студентів
    if (!(fp=fopen("fp.dat","rb"))) cerr<<"File do not exist!";
    else
    {
        fread(&predmet,sizeof(subject),1,fp);
        int k=0;
        while(!feof(fp))
        {
            fond_prep+=predmet.prepod.salary;
            for(j=0;j<predmet.stnum;j++) //перегляд студентів
                //по даній дисципліні
            {
                //копіювання масиву студентів
                tmp[k]=predmet.student[j];
                k++;
            }
        }
    }
}

```



```

        fread(&predmet,sizeof(subject),1,fp);
    }
    fclose(fp);          //закриття файлу
    for (i=0;i<k;i++)    //перегляд всього масиву студентів
    {
        int f=0;        //прапорець обробки даних по студенту
        for (j=0;j<i;j++)//перегляд студентів спочатку масиву до даного запису
            if ((strcmp(tmp[i].fio,tmp[j].fio)==0))
                f=1;      //студент вже зустрічався в масиві
        if (f==0)        //студент іще не зустрічався в масиві
            fond_stip+=tmp[i].stip;
    }
    cout<<"salary found ="<<fond_prep<<endl;
    cout<<"scholarship found ="<<fond_stip<<endl;
    if (fond_prep-fond_stip>0)
        cout<<"salary found is bigger "<<endl;
    else
        cout<<"salary found is smaller "<<endl;
}
getch();                //чекати натискання клавіші
}
//-----головна програма-----
void main()
{
    while (1)
    {
        system("cls");
        cout<<"          MENU  \n"<<endl;
        cout<<"1. Database input by keyboard."<<endl;
        cout<<"2. Read data from file."<<endl;
        cout<<"3. View teacher's database."<<endl;;
        cout<<"4. View student's database."<<endl;
        cout<<"5. List of subjects with the highest average ball."<<endl;
        cout<<"6. Scholarship."<<endl;
        cout<<"7. Scholarship found."<<endl;
        cout<<"8. Exit"<<endl;
    }
}

```

```

cout<<endl;
int key;           //номер пункту меню
cout<< "Choosen menu item:";
cin>>key;
system("cls");
switch(key)        // вибір функції реалізації меню
{
    case 1:  {input();break;}
    case 2:  {read_from_file();break;}
    case 3:  {list_prepod();break;}
    case 4:  {list_student();break;}
    case 5:  {maxbal();break;}
    case 6:  {stipendia();break;}
    case 7:  {fond();break;}
    case 8:  {exit (0); break;}
    default: {cout<<"Try another time...">getch();break;}
}
}
}

```

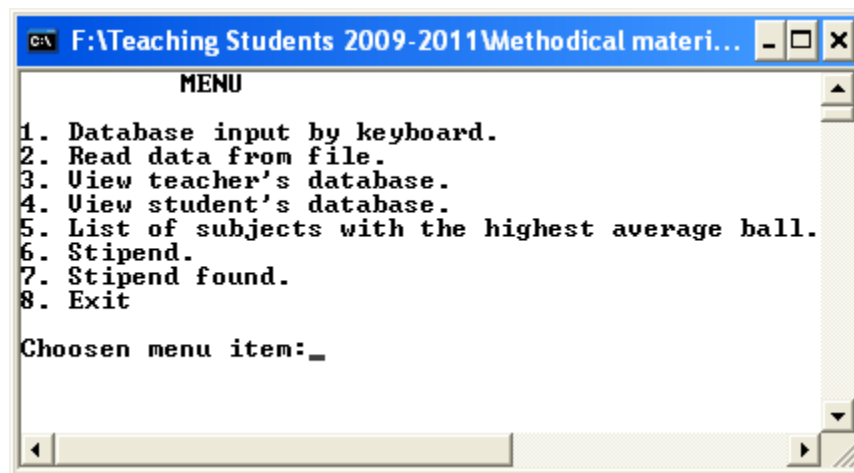


Рис. 3. Результати роботи програми обробки структур і файлів. Зображення меню

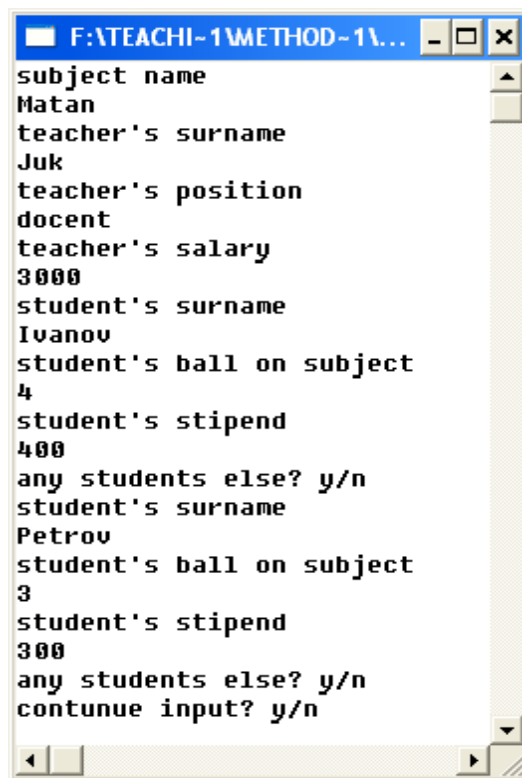


Рис. 4. Результати роботи програми обробки структур і файлів. Створення файлу

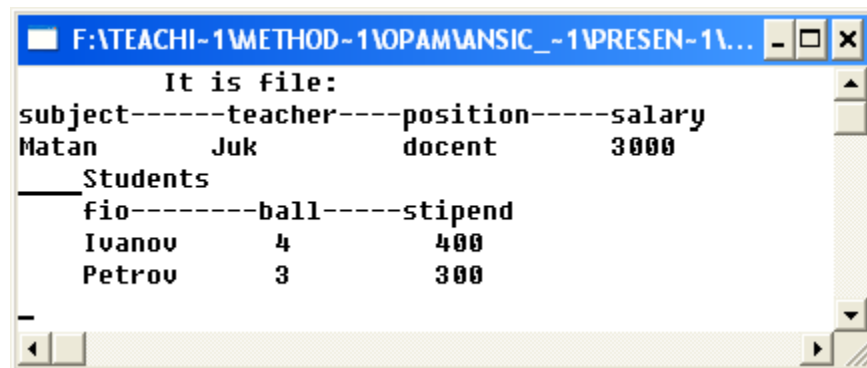


Рис. 5. Результати роботи програми обробки структур і файлів. Читання з файлу

Варіанти завдань

1. Створити масив структур. Кожна структура складається з таких елементів: факультет, курс, група, предмет, студент. Для предмета задається назва та кількість кредитів. Для студента задається прізвище та екзаменаційні оцінки. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) прізвища студентів, які мають дві та більше двійок за сесію і вилучити їх; 2) факультет, який на першому курсі має найбільшу кількість відмінників; 3) курс, на якому виключено найбільшу кількість студентів.
2. Створити масив структур. Кожна структура складається з таких елементів: фірма, робітник, посада. Для робітника задається прізвище та заробітна платня. Для посади задається кількість вакантних місць, вимоги до претендентів на кожну посаду, Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити фірми з найбільшою кількістю вакансій на задану посаду; 2) вивести список вимог до кандидатів на задану посаду

по кожній фірмі; 3) створити новий масив із структурою: вакантна посада, заробітна платня. Вакансії вибирати з початкового масиву, виключаючи повторення однакових вакантних посад. На випадок різної платні на однакових посадах у різних фірмах включити у масив середню заробітну платню.

3. Створити масив структур. Кожна структура складається з таких елементів: факультет і студент. Для факультету задати його назву, курс і масив груп. Для студента задати прізвище та масив екзаменаційних оцінок. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити факультет і курс з максимальною кількістю відмінників; 2) вивести список відмінників; 3) знайти групу, де немає двієчників.

4. Створити масив структур. Кожна структура складається з таких елементів: інститут та студент. Для інституту задати його назву і масив факультетів. Для студента задати курс, групу, прізвище та середній бал. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) факультет і курс, на якому середній бал не менше 3.5; 2) прізвища студентів із , що навчаються на вказаному курсі та групі; 3) факультет і групу, де найбільше відмінників.

5. Створити масив структур. Кожна структура складається з таких елементів: місто, інститут, факультет. Для інституту задається план прийому на перший курс. Для факультету задається список спеціальностей. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) список спеціальностей, що містять у своїй назві слова "комп'ютерний" або "автоматизований"; 2) факультети й інститути, де є задана спеціальність; 3) факультет, інститут і місто, де на вказану користувачем спеціальність виділено найбільше місць.

6. Створити масив структур, кожна з яких складається з таких елементів: факультет, курс, студент. Для факультету задати його назву та номер групи. Для студента задати прізвище студентів, екзаменаційні оцінки. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити прізвища студентів, групу і факультет, де середній бал студентів 4.5; 2) вилучити прізвища студентів першого курсу, які мають три двійки; 3) відсортувати назви факультетів за умови зростання успішності студентів.

7. Створити масив структур. Кожна структура складається з таких елементів: факультет, курс, група, предмет, студент. Для предмета задається назва та кількість кредитів. Для студента задається прізвище та екзаменаційні оцінки. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) прізвища студентів, які мають дві та більше двійок за сесію і вилучити їх; 2) факультет, який на першому курсі має найбільшу кількість відмінників; 3) курс, на якому виключено найбільшу кількість студентів.

8. Створити масив структур. Кожна структура складається з таких елементів: фірма, робітник, посада. Для робітника задається прізвище та заробітна платня. Для посади задається кількість вакантних місць, вимоги до претендентів на кожну посаду. Створений масив записати до бінарного файлу. Передбачити можливість

доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити фірми з найбільшою кількістю вакансій на задану посаду; 2) вивести список вимог до кандидатів на задану посаду по кожній фірмі; 3) створити новий масив із структурою: вакантна посада, заробітна платня. Вакансії вибирати з початкового масиву, виключаючи повторення однакових вакантних посад. На випадок різної платні на однакових посадах у різних фірмах включити у масив середню заробітну платню.

9. Створити масив структур. Кожна структура складається з таких елементів: факультет і студент. Для факультету задати його назву, курс і масив груп. Для студента задати прізвище та масив екзаменаційних оцінок. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити факультет і курс з максимальною кількістю відмінників; 2) вивести список відмінників; 3) знайти групу, де немає двічників.

10. Створити масив структур. Кожна структура складається з таких елементів: інститут та студент. Для інституту задати його назву і масив факультетів. Для студента задати курс, групу, прізвище та середній бал. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) факультет і курс, на якому середній бал не менше 3.5; 2) прізвища студентів, що навчаються на вказаному курсі та групі; 3) факультет і групу, де найбільше відмінників.

11. Створити масив структур. Кожна структура складається з таких елементів: місто, інститут, факультет. Для інституту задається план прийому на перший курс. Для факультету задається список спеціальностей. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) список спеціальностей, що містять у своїй назві слова "комп'ютерний" або "автоматизований"; 2) факультети й інститути, де є задана спеціальність; 3) факультет, інститут і місто, де на вказану користувачем спеціальність виділено найбільше місць.

12. Створити масив структур, кожна з яких складається з таких елементів: факультет, курс, студент. Для факультету задати його назву та номер групи. Для студента задати прізвище студентів, екзаменаційні оцінки. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити прізвища студентів, групу і факультет, де середній бал студентів 4.5; 2) вилучити прізвища студентів першого курсу, які мають три двійки; 3) відсортувати назви факультетів за умови зростання успішності студентів.

13. Створити масив структур. Кожна структура складається з таких елементів: марка автомобіля, тип (вантажний або легковий), номерний знак, строк служби. Для легкових вказується колір, для вантажних - вантажність. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити номерні знаки автомобілів, вантажність котрих не менше за задану; 2) вивести марки автомобілів заданого типу і вказаного кольору; 3) вилучити з файлу відомості про автомобілі, строк служби яких перевищує п'ять років.

14. Створити масив структур. Кожна структура складається з таких елементів: абонент і телефон. Для

абонента задається прізвище, адреса, заборгованість по оплаті Для телефону задається номер і оператор (Life, Київстар тощо). Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити прізвища абонентів, що мають заборгованість по оплаті більше заданої користувачем; 2) вилучити прізвища абонентів, адреса яких змінилася; 3) замінити номер телефону заданого абонента.

15. Створити масив структур. Кожна структура складається з таких елементів: фірма, товар. Для фірми задається назва, кількість товарів. Для товару задається його найменування, вартість, термін поставки. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити фірми, що постачають заданий товар у дводенний строк; 2) визначити назву товару в заданій фірмі, вартість якого не перевищує заданого значення; 3) впорядкувати масив за умови зростання термінів постачання товару.

16. Створити масив структур. Кожна структура складається з таких елементів: інститут, факультет. Для інституту задати назву та план прийому на перший курс. Для факультету задати список спеціальностей та прохідний бал. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) скласти список спеціальностей по факультетах із вказаним прохідним балом; 2) впорядкувати масив за прохідним балом і планом прийому; 3) визначити інститут із вказаним середнім прохідним балом.

17. Створити масив структур. Кожна структура складається з таких елементів: літак і рейс. Для літака задати тип, кількість вільних місць, вартість квитка. Для рейсу задати номер, дата польоту. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) скласти список номерів авіаційних рейсів, на які є вільні місця; 2) вивести типи літаків і відповідні номери авіаційних рейсів на задану дату; 3) упорядкувати масив за умови зростання вартості квитків по кожній даті.

18. Створити масив структур. Кожна структура складається з таких елементів: напрям польоту та літак. Для напрямку польоту задати льотні дні на тиждень і номер авіаційного рейсу. Для літака задати тип, кількість посадкових місць, вартість квитка. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити номери авіаційних рейсів заданого напрямку з мінімальною вартістю квитків; 2) визначити типи літаків і сумарну кількість посадкових місць у заданому напрямку по днях тижня; 3) вилучити з масиву відомості про рейси, якщо кількість польотів на тиждень менше двох.

19. Створити масив структур. Кожна структура складається з таких елементів : виробник, товар, склад. Для виробника задати назву та адресу фірми. Для товару – назву товару, вартість, об'єм партії товару, що постачається. Для складу задати термін прийняття товару та термін його зберігання. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані у файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) виробника, який постачає заданий товар в заданий термін.;

2) товар, що може зберігатися найдовше на складі; 3) товар об'єм партії якого найбільший.

20. Створити масив структур. Кожна структура складається з таких елементів: абонент та телефон. Для абонента задати його прізвище та адресу. Для телефону задати його номер, вид оплати (щохвилинна чи абонементна), вартість оплати. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати такі операції: 1) доповнити масив прізвищами абонентів, номери телефонів яких починаються на задані цифри (імітувати введення нової АТС); 2) видати список абонентів з щохвилинною оплатою, сума оплати у яких нижче заданого рівня; 3) визначити номер телефону абонента за заданою адресою.

21. Створити масив структур. Кожна структура складається з таких елементів: напої та шоколадні вироби; їх калорійність, вартість. Для напоїв вказати міцність, для шоколадних виробів - вагу, начинку. Створений масив структур записати до файлу. Передбачити операції додавання до файлу записів і їх редагування. Реалізувати запити: 1) визначити список шоколадних виробів, що мають максимальну вагу і задану начинку; 2) вилучити з масиву інформацію про алкогольні напої; 3) скласти меню з напоїв і шоколадних виробів, що відповідають заданій калорійності та вартості.

22. Створити масив структур. Кожна структура складається з таких елементів: споживач і продукт. Для споживача задати його прибуток. Для продукту задати ціну, рівень корисності продукту (функцію корисності задати у довільному вигляді). Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) продукти заданого рівня корисності, які може придбати споживач при заданому прибутку; 2) сумарну вартість продуктів з максимальним рівнем корисності; 3) споживачів, прибутків яких не вистачає для придбання продуктів за заданою ціною і рівнем корисності.

23. Створити масив структур. Кожна структура складається з таких елементів: інститут, факультет. Для інституту задати його назву, місто, де він розташований, план прийому на перший курс. Для факультету задати його назву, масив спеціальностей, прохідний бал. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) список міст, де знаходяться політехнічні інститути; 2) інститути і факультети, де приймають студентів на перший курс за заданою користувачем спеціальністю; 3) факультет, інститут і місто, де на вказану користувачем спеціальність існує найвищий прохідний бал.

24. Створити масив структур. Кожна структура складається з таких елементів: факультет, прізвища студентів, прізвища викладачів. Для студентів вказують стипендію і середній бал. Для викладачів - посаду і заробітну плату. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) список студентів по факультетах з найнижчим середнім балом і стипендією; 2) кількість викладачів-доцентів на заданому факультеті; 3) сумарний фонд стипендії і зарплати на заданому факультеті.

25. Створити масив структур. Кожна структура складається з таких елементів: фірма, товар, що продає фірма.

Для фірми задати назву, регіон, в якому товар продається. Для товару задати вид (комп'ютери і програмне забезпечення), вартість продажу, термін постачання. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) список комп'ютерів, що продаються у заданому регіоні заданою фірмою; 2) вартість проданого програмного забезпечення у задані терміни; 3) найрентабельніші фірми (з найбільшою вартістю продажів).

26. Створити масив структур. Кожна структура складається з таких елементів: прізвище студента, курс, середній бал. Для студентів контрактної форми навчання вказується вартість контракту, а для студентів, що навчаються за бюджетною формою - розмір стипендії. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати такі операції: 1) визначити прізвища студентів із середнім балом вище четвірки, надрукувати окремо студентів контрактної та бюджетної форм навчання; 2) вилучити дані про студентів бюджетної форми з файлу, якщо їх середній бал нижче трійки; 3) вивести список студентів контрактної форми і суму їх контрактів.

27. Створити масив структур. Кожна структура складається з таких елементів: книги, журнали. Для книжок задають назву, прізвище автора, рік видання. Для журналів - назву, номер, рік видання, назву статті, автора. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) вивести авторів, які друкувалися у вказаному році та їх праці; 2) визначити журнал, рік видання вказаної статті; 3) вивести кількість статей заданого автора і журнали, у яких він друкувався.

28. Створити масив структур. Кожна структура складається з таких елементів: прізвище викладача, дисципліна, прізвища студентів, що вивчають конкретну дисципліну. Для викладача задають посаду, заробітну плату, для студента - середній бал і стипендію. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати такі запити: 1) визначити викладачів і дисципліни з середніми балами менше заданого; 2) вилучити студентів, середній бал яких нижче за трійку; 3) визначити стипендіальний фонд студентів (кількість студентів, що отримують стипендію, помножену на розмір стипендії), які вивчають задану дисципліну.

29. Створити масив структур. Кожна структура складається з таких елементів: книги, читачі. Для книги задати її назву, прізвище автора, рік видання. Для читача – прізвище, адреса, термін повернення книги. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити, визначивши: 1) прізвища читачів, які читають задані книги; 2) книгу заданого автора, що має повернутися в заданий термін; 3) прізвища читачів-боржників (тих, хто в заданий термін книги не повернули).

30. Створити масив структур. Кожна структура складається з таких елементів: пасажир, багаж. Для пасажир задати його прізвище та вагу багажу. Для багажу задати кількість та вартість речей. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити

дані з файлу. Реалізувати запити, визначивши: 1) пасажирів з найбільшою вагою багажу; 2) багаж заданого пасажирів; 3) упорядкувати масив структур за умови зростання вартості багажу.

31. Створити масив структур. Кожна структура складається з таких елементів: фірма, продукт, що виробляє фірма. Для фірми задати її назву, назву міста. Для продукту задати його назву, кількість, ціну одиниці продукту. Створений масив записати до бінарного файлу. Передбачити можливість доповнити файл, замінити дані в файлі, видалити дані з файлу. Реалізувати запити: 1) визначити місцезнаходження фірми, що виробляє заданий продукт; 2) визначити сумарну вартість продуктів, що виробляються у даному місті; 3) упорядкувати список продуктів за ціною для даного міста.

Контрольні питання

1. Означити поняття логічного та фізичного файлу.
2. Чим файли і масиви схожі? Чим вони відрізняються?
3. Чим відрізняються бінарні файли від текстових?
4. Чим зумовлена потреба у відкритті та закритті файлів?
5. Що таке файловий покажчик? Які стандартні функції змінюють його значення?
6. Як здійснюється навігація по файлу?
7. Як визначається кінець файлу у програмі? Як визначається кінець фізичного файлу?
8. Чим відрізняється послідовний доступ до компонентів файлу від прямого доступу?

Лабораторна роботи №15 "Файли в C++"

1. Дано файл *f*, компоненти якого є дійсними числами. Знайти найбільше зі значень модулів компонент непарними номерами.
2. Дано файл *f*, компоненти якого є дійсними числами. Знайти різницю першої і останньої компоненти файлу.
3. Дано файл *f*, компоненти якого є цілими числами. Знайти кількість подвоєних непарних чисел серед компонентів.
4. Дано символічний файл *f*. У файлі *f* не менше двох компонент. Визначити, чи є два перших символу файлу цифрами. Якщо так, то встановити, чи є число, утворене цими цифрами, парним.
5. Дано символічний файл *f*. Отримати, файл *g*, утворений з файлу *f* заміною всіх його прописних (великих) букв однойменними рядковими (малими).
6. Дано файл *f*, компоненти якого є цілими числами. Записати в файл *g* всі парні числа файлу *f*, а в *p* всі непарні. Порядок проходження чисел зберігається.
7. Дано файл *f*, компоненти якого є цілими числами. Отримати файл *g*, утворений з файлу *f* видаленням повторних входжень одного і того ж числа.
8. Дано файл *f*, компоненти якого є цілі числа. Ніяка з компонент файлу не дорівнює нулю. Файл *f* містить стільки ж від'ємних чисел, скільки і додатних. Використовуючи допоміжний файл *h* переписати всі компоненти файлу *f* в файл *g* так, щоб у файлі *g* не було двох сусідніх чисел з одним знаком.
9. Дано символічний файл *f*. Підрахувати число входжень в файл поєднань *ab*.
10. Дано символічний файл *f*. Визначити чи входить в файл поєднання *abcd*.
11. Дані символічні файли *f* і *g*. Записати в файл *h* всі початкові значення компонент, які збігаються у файлах *f* та *g*.
12. Дано символічний файл *f*. Видалити з файлу все однобуквені слова і зайві прогалини. Результати записати в файл *g*.
13. Дано файл *f*, компоненти якого є цілими числами. Ніяка з компонент файлу не дорівнює нулю. Файл *f* містить стільки ж від'ємних компонент, скільки і додатних. Використовуючи допоміжний файл *h* переписати всі компоненти файлу *f* в файл *g* так, щоб у файлі *g* спочатку йшли додатні, потім від'ємні.
14. Дано файл *f*, компоненти якого є цілі числа. Ніяка з компонент файлу не дорівнює нулю. Числа в файлі йдуть в такому порядку: десять від'ємні, десять додатних, десять від'ємних, десять додатних і т.д. Переписати компоненти файлу *f* в *g* так, щоб у файлі *g* числа йшли в наступному порядку: п'ять від'ємних, п'ять додатних і т.д.
15. Дано символічний файл *f*. Вважаючи, що кількість символів в слові не перевищує 20, визначити, скільки в файлі є слів, що складаються з одного, двох, трьох і т.д. символів.
16. Дано символічний файл, який містить відомості про студентів КПІ, записані за таким зразком: прізвище ім'я по батькові, прізвище ім'я по батькові... Записати ці відомості в файл *g*, використовуючи зразки: ім'я по батькові прізвище, ім'я по батькові прізвище...

17. Дано символний файл, який містить відомості про студентів групи ДА-81, записані за таким зразком: ім'я по батькові прізвище, ім'я по батькові прізвище... Записати ці відомості в файл g, використовуючи зразки: прізвище ім'я по батькові, прізвище ім'я по батькові...
18. Дано файл f, який містить номери телефонів групи ДА-82. Вказується прізвище студента, його ініціали та номер телефону. Знайти телефон студента по його прізвищу і ініціалам.
19. Дано текстовий файл. Отримати всі його рядки, які містять більше 60 символів.
20. Дано символний файл f. Вважаючи, що кількість символів в слові не перевищує 20, отримати гістограму довжин всіх слів файлу f.
21. Заданий текстовий файл F, рядок букв S. Отримати всі рядки файлу, які містять в собі рядок S.
22. Заданий текстовий файл F. Переписати компоненти файлу F в файл H, вставляючи в початок символ пропуску. Порядок компонент повинен зберігатися.
23. У заданому текстовому файлі виділити найдовший рядок файлу, якщо їх декілька, то виділити якусь з них.
24. Заданий текстовий файл F, який містить програму на мові C. Перевірити текст на різну кількість відкритих і закритих дужок, вважаючи, що кожен оператор в програмі займає не більше одного рядка в файлі F.
25. Заданий файл, який містить телефонні номери студентів кафедри. Вказується прізвище студента, його ініціали та номер телефону. Ввести прізвище та ім'я з терміналу. Знайти телефон по введених даних.
26. Заданий файл містить різні дати, кожна з яких - це число, місяць і рік. Знайти всі весняні дати.
27. Пряма на площині задається рівністю $Ax + By + C = 0$. Нехай F-файл, який містить коефіцієнти декількох прямих. Переписати з файлу F в файл G коефіцієнти прямих, які паралельні першій прямій, які задані в файлі F.
28. Заданий текстовий файл. Переписати файл, закінчуючи кожен рядок символом «-», якщо рядок в заданому файлі закінчується на символ, відмінний від символів пробіл, коми або крапки.
29. У заданому текстовому файлі виділити всі рядки, які містять більше ніж 60 символів.
30. Заданий текстовий файл F. Переписати в файл G всі компоненти файлу, замінивши в них символ 0 на 1 і навпаки.
31. Отримати копію заданого файлу F, переписавши його в файл G (обидва файли символні). У новому файлі на початок кожного рядка записати відповідний номер рядка.
32. Задані символні файли F і G. Визначити збігаються компоненти файлу F з компонентами файлу G. Якщо ні, то визначити номер першої компоненти, починаючи з якої файли F і G відрізняються між собою.
33. Заданий файл F, компоненти якого натуральні числа. Кількість чисел у файлі кратна 4. Перші два числа з кожних 4-х задають координати лівого верхнього кута прямокутника, наступні два - координати правого нижнього кута. Побудувати прямокутники, які задані у файлі F.

Лабораторна робота № 16 «Директиви препроцесора»

Мета роботи: Вивчити особливості використання директив препроцесора, навчитися застосовувати директиви препроцесора під час налагодження програми.

Теоретичні відомості

У мові C/C++ використовують директиви включення, макровизначення, умовної компіляції, виведення повідомлення про помилку та ін.:

`#include ім'я файлу`

`#define ім'я значення`

`#if вираз`

`#else`

`#elif вираз`

`#endif`

`#error повідомлення_файлу та інш.`

Директиви включення використовують для включення до програмного коду бібліотечних файлів, які розміщуються за адресою `.../vc/include` та мають шаблон імені `*.h` або без розширення імені, наприклад, `iostream`.

Формат директиви:

`#include <ім'я файлу>`

Для підключення файлів користувача до головної програми використовується такий формат директиви:

`#include "ім'я файлу"`

Макровизначення (макроси) – це символні константи та операції, які позначаються символними рядками. Вони забезпечують підстановку значення замість імені, визначеного у макросі, при звертанні до цього імені у функціях програми.

Формат макросу:

`#define < id1 > [(< id2 >, ...)] < token string >`

де `id1` - ідентифікатор макросу; `id2` - значення ідентифікатора; `token string` - рядок маркерів, який містить ключові слова, константи, оператори.

Вимоги до макросів:

- ім'я макросу пишуть, як правило, заголовними літерами для відмінності їх від функцій;
- ім'я макросу від його тіла відділяється хоча б одним пропуском;
- кінцем макросу є символ нового рядка (натиснення клавіші ENTER);
- параметри макросу записують у круглих дужках () як в імені, так і тілі макросу;
- макроси можуть бути вкладеними;
- для продовження макросу на новому рядку використовується символ продовження макросу `'/'` на попередньому рядку;

- існують глобальні ідентифікатори (DATE, FILE, LINE, TIME, STDC), які не можна використати у директивах #define або #undef;
- у макросах з параметром не використовуються складні оператори, такі як оператори циклу, розгалуження, перемикачання та ін.

Директиви умовної компіляції дозволяють вибірково компілювати частини коду залежно від значення деякого константного виразу або ідентифікатора.

Конструкція 1

```
#if <вираз>    //якщо вираз істинний, то компілюються рядки, які йдуть за #if,
#else          //інакше компілюються рядки, розміщені після #else до #endif
#endif        //кінець умовної компіляції
```

Конструкція 2

```
#if <вираз1>    //якщо вираз1 істинний, то компілюються рядки, записані після #if
#elif <вираз2>  //компілюються рядки після #elif, якщо вираз1 хибний, а вираз2 істинний
#else          //інакше компілюються рядки, записані після #else
#endif        //кінець умовної компіляції
```

В директивах #if, #elif вираз має бути сталим (наприклад, #if 1).

Директива повідомлення про помилку #error викликає під час компіляції повідомлення про помилку. За звичай її використовують тоді, коли не було визначено певний макрос. Може містити ідентифікатори макросів, які розширюються препроцесором.

Формат директиви:

```
#error <повідомлення>
```

Макроси обробки змінного числа параметрів.

Прототипи макросів містяться у заголовних файлах stdarg.h, vararg.h. Використовуються при оголошенні функцій із змінною кількістю параметрів, яка вказується крапками (...) у заголовку функції.

Формати макросів:

```
type va_arg(va_list arg_ptr, type);
void va_end(va_list arg_ptr);
void va_start(va_list arg_ptr); // (UNIX version)
void va_start(va_list arg_ptr, prev_param); // (ANSI version)
```

де type – тип аргументу, який відновлюється; arg_ptr – покажчик на список аргументів; prev_param – параметр, який передує першому необов'язковому аргументу (ANSI тільки).

Макрос va_arg повертає поточний аргумент; макроси va_start і va_end не повертають значення.

Послідовність використання цих макросів чітко визначена:

- оголошується змінна типу va_list, яка ініціалізується значенням макросу va_start();
- використовується va_arg() для доступу до одного або декількох параметрів необхідного типу;

- використовується `va_end()` для визначення кінця обробки параметрів;
- не припускається використання параметрів типів: `char`, `unsigned char`, `float`.

Макроси класифікації символів ASCII.

Прототипи макросів містяться у заголовному файлі `ctype.h`: `isalnum(int)`, `isalpha(int)`, `isascii(int)`, `isctrl(int)`, `isdigit(int)`, `ispunct(int)`, `isspace(int)`, `islower(int)`, `isupper(int)`.

Приклади програм

Приклад 1

Обчислити площу кола, знайти максимум з двох чисел, використовуючи макроси піднесення до квадрата, виведення результатів, знаходження максимального значення. За допомогою директив умовної компіляції активізувати різні частини програми.

Алгоритм задачі

1. Якщо значення константи менше 10, то виконати дії:
 - 1.1. Увести три цілих числа;
 - 1.2. Вивести результат обчислення квадрата першого числа;
 - 1.3. Вивести результат визначення квадрата суми другого числа та константи N;
 - 1.4. Вивести результат знаходження максимуму між першим і другим квадратами чисел.
2. Якщо значення константи більше 10, то виконати дії:
 - 2.1. Увести радіус кола;
 - 2.2. Обчислити площу кола;
 - 2.3. Вивести визначений результат.
3. Запит на повторне виконання програми.
4. Якщо дано згоду, то повторити п.п.1 - 3, інакше - кінець розрахунків.

Приклад коду

*/*Обчислити площу кола, знайти максимум з двох чисел, використовуючи макроси піднесення до квадрата, виведення результатів, знаходження максимального значення. За допомогою директив умовної компіляції активізувати різні частини програми.*/*

```
#include <stdio.h>
#include <conio.h>

#define N 5                //директива препроцесора – макровизначення
#define SQR(x) (x) * (x)  //макрос з параметром – визначення
                           //квадрата числа
                           //макрос визначення найбільшого з двох чисел
#define MAX(y, z) ((y)>(z))?(y):(z) //макрос як умовний вираз
                           //виведення повідомлення та значення цілого числа
#define KTPR(w) puts("control output"); \
```

```

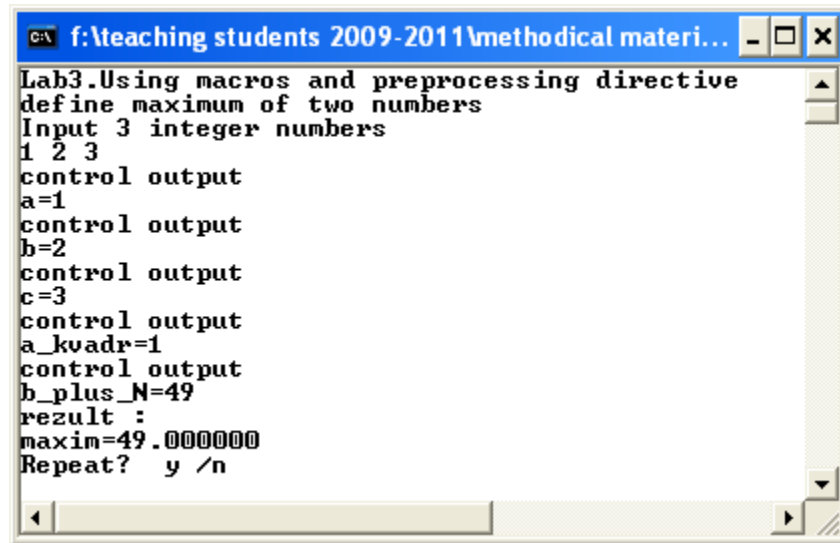
printf("#w"="%d\n",w) //використовується символ продовження макроса '\
                        //символ '#' використовується для об'єднання рядків
#define REZPR(w) puts ("rezult :");\
                        printf (#w"=%f\n",(float)w) //виведення повідомлення
                                                //виведення значення дійсного числа
#define SQCIRLE(r) (3.14*SQR(r)) //площа кола
//=====основна програма=====
void main()
{
    puts("Lab3.Using macros and preprocessing directive");
    int a, b, c, //змінні для розрахунків
        b_plus_N, //допоміжна змінна
        a_kvadr, //змінна для визначення квадрата числа
        maxim; //максимальне значення серед трьох чисел
    char ch; //відповідь на запитання про продовження програми
    float rad, s; //радіус і площа кола
    do
    {
        #if N<10 //якщо умова істинна, то розрахувати максимум з трьох чисел
            puts("define maximum of two numbers");
            puts("Input 3 integer numbers");
            scanf ("%d",&a); //введення числа
            KTPR(a); //макрос виведення числа
            scanf ("%d",&b);
            KTPR(b);
            scanf ("%d",&c);
            KTPR(c);
            a_kvadr=SQR(a); //макрос визначення квадрата числа
            KTPR(a_kvadr);
            b_plus_N=SQR(b+N);
            KTPR(b_plus_N);
            maxim=MAX(a_kvadr,b_plus_N); //пошук найбільшого з двох чисел
            maxim=MAX(maxim,c);
            REZPR(maxim); //макрос виведення результату
        #else //компілювати іншу частину програми, якщо задати N >=10
            puts("define area of a circle");

```

```

puts("Input radius of circle");
scanf("%f", &rad);
s = SQCIRCLE(rad);
REZPR (s);
#endif                                     //завершення умовної компіляції
puts ("Repeat?  y /n ");    ch=getch();
} while (ch=='y');
}

```



```

Lab3.Using macros and preprocessing directive
define maximum of two numbers
Input 3 integer numbers
1 2 3
control output
a=1
control output
b=2
control output
c=3
control output
a_kvadr=1
control output
b_plus_N=49
result :
maxim=49.000000
Repeat?  y /n

```

Рис. 6. Результати роботи програми застосування макросів та директив умовної компіляції

Приклад 2

```

/* The program below illustrates passing a variable number of arguments using
the following macros: va_start, va_arg, va_end, va_list */
#include <stdio.h>
#include <conio.h>
#include <stdarg.h>
float average( int first, ... );
#include <varargs.h>
float average( va_list );
//===== головна програма =====
int main( void )
{
    puts("function with a variable number of arguments");
    /* Call with 3 integers (-1 is used as terminator). */
    printf( "Average (2, 3, 4, -1) is: %f\n", average( 2, 3, 4, -1 ) );
    /* Call with 4 integers. */
    printf( "Average (5, 7, 9, 12, -1) is: %f\n", average( 5, 7, 9, 12, -1 ) );
}

```



```

/* Call with just -1 terminator. */
printf( "Average (-1 ) is: %f\n", average( -1 ) );

while(!kbhit()); //затримка зображення поки не натиснута
                //будь-яка клавіша
}

//===== Returns the average of a variable list of integers=====
float average( int first, ... )
{ //first - перший аргумент
    int count = 0;
    int sum = 0, i = first;
    va_list marker;
    va_start( marker, first );      /* Initialize variable arguments. */
    while( i != -1 )
    {
        sum += i; //накопичення суми аргументів
        count++; //розрахунок кількості аргументів
        i = va_arg( marker, int); //визначення наступного аргументу
    }
    va_end( marker );                /* Reset variable arguments.      */
    float rez = (float) sum / count;
    return ( sum ? rez : 0 );
}

```

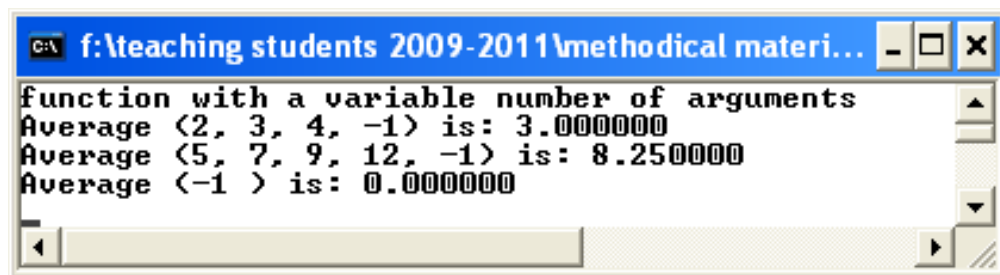


Рис. 7. Результати роботи програми застосування макросів обробки змінного числа параметрів

Варіанти завдань

1. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} \max(x^2 + y + z, xyz), & 0 < x < 5, \\ \min(x^2 + (y + z)^2, xy), & x \leq 0, x \geq 5. \end{cases}$$

2. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} \max^2(x + y, x + z), & 1 \leq a < 10, \\ \min(x^2, 5 * (y - z)), & a < 1, a \geq 10. \end{cases}$$

3. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} (x + y) \text{ and } (yz) \text{ or } (\text{not } z), & a = 0, \\ \text{not}((0 \text{ or } x) \text{ and } (x - y)), & a \neq 0. \end{cases}$$

4. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} \max(|x| + |y| + |z|, x^3), & 0 < a < 3, \\ \min(2x + y(2y - z)^2), & a > 3. \end{cases}$$

5. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} \min(x, z) * \max(x + z, x * z), & 5 < a < 10, \\ \max^2(x - z, z), & a \leq 5, z \geq 10. \end{cases}$$

6. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем.

$$w = \begin{cases} \max((x - y)^2, |y - x|), & a = 1, \\ \min(x + y^2 \sqrt{x + y}), & a \neq 1. \end{cases}$$

7. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції $\max()$, $\min()$, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем. Самостійно

визначити умови вибору.

$$w = \begin{cases} \sqrt{x-y} + \max((x-z)^2, (x-y)), \\ \max(xy, yz) + \min(x-y, y-z). \end{cases}$$

8. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції `max()`, `min()`, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем. Самостійно визначити умови вибору.

$$w = \begin{cases} ((x \text{ and } y) \text{ and } (x \text{ or } z)) \text{ and } (\text{not } x \text{ xor } y), \\ ((x \text{ or } y) \text{ and } (z \text{ or } y)) \text{ and } (z \text{ or } x). \end{cases}$$

9. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції `max()`, `min()`, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем. Самостійно визначити умови вибору.

$$f = \begin{cases} \max(x^3, (x+y)^2) * \min(x-y, y-z), \\ \min(|x-z|, |x+z|). \end{cases}$$

10. Використовуючи директиви умовної компіляції, макровизначення, включення власних заголовних файлів для необхідних оголошень та інші директиви, скласти програму обчислення таких виразів. Функції `max()`, `min()`, визначення модуля виразів, виведення результатів, логічних виразів, піднесення до степені, визначення коренів задати у вигляді макросів. Значення змінних задаються користувачем. Самостійно визначити умови вибору.

$$f = \begin{cases} \max^2(x, y, z) - 2 * \min(x, y, z), \\ \sin^2(x) + \max(x, y, z) / \min(x, y, z). \end{cases}$$

11. Використовуючи макроси обробки змінного числа параметрів, скласти програму введення-виведення та визначення суми довільної кількості чисел різних типів.

12. Використовуючи один і той самий ідентифікатор, оголосити масиви чисел і символів. Для масиву чисел визначити кількість повторень кожного числа. Для масиву символів знайти позицію і значення символу за заданим кодом символу. Передбачити умовну компіляцію. Початок і кінець функцій та блоків оформити у стилі мови Pascal, використовуючи `begin` та `end`.

13. Використовуючи макроси класифікації символів ASCII, визначити кількість кожних типів символів (алфавітні, цифрові, керуючий, знаки пунктуації та ін.) у процесі читання текстового файлу. Виділити цифрові символи із заданого користувачем діапазону за допомогою макросів `assert()`.

14. Використовуючи макроси, задати три числа. Визначити площу трикутника, якщо виконується умова його

існування, інакше обчислити площу трапеції, задавши її основу та висоту за умови існування трапеції, інакше видати повідомлення про помилку.

15. Довести, що катети і висота, проведена до гіпотенузи, пов'язані співвідношенням: $1/H^2 = 1/a^2 + 1/b^2$, де H - висота, a , b - катети. Попередньо перевірити, чи є трикутник прямокутним. Необхідні обчислення і початкові дані задати самостійно.

16. Організувати діагностику роботи програми за допомогою наперед визначених макросів FILE, LINE та макросів виведення повідомлень.

17. Використовуючи макроси умовної компіляції та оператор defined, перевірити правильність встановлених опцій компілятора і видати відповідні повідомлення.

18. Довести, що для будь-якого трикутника ABC виконується нерівність: $\cos A + \cos B + \cos C \leq 3/2$

19. Перевірити введені числа на парність і додатність. Оформити у вигляді макросів визначення парності, додатності чисел, виведення діагностичних повідомлень. Шляхом умовної компіляції поділити процеси визначення парності та додатності чисел.

20. Використовуючи допомогою наперед визначені макроси DATE, FILE, LINE, STDC, TIME, вивести повідомлення про дату, час, ім'я файлу тощо. Передбачити повідомлення про програмні помилки. Здійснити передачу параметрів у програму і блокування їх передачі за допомогою директиви #pragma.

21. Скласти довідник операцій над множинами: об'єднання, перетинання, входження, рівності, різниці, доповнення тощо, задавши необхідні визначення у вигляді макросів і надрукувавши їх.

22. Використовуючи макроси класифікації символів ASCII, визначити алфавітні та цифрові символи у процесі введення рядка з клавіатури. З цифрових символів побудувати матрицю вимірності, яку визначають за таким правилом: кількість рядків дорівнює значенню першого цифрового символу, відмінного від нуля, кількість стовпчиків дорівнює коду першого алфавітного символу, зменшеного у n разів. Необхідні константи задати макросами.

23. Використовуючи макроси обробки змінної кількості параметрів, скласти програму обчислення найбільшого спільного дільника довільної кількості цілих чисел.

24. Використовуючи макроси обробки змінної кількості параметрів, скласти програму генерації повідомлень про помилки по її коду за допомогою передачі кожного слова як окремого рядкового параметра. Відповідність коду помилки і повідомлення з контексту задачі визначає користувач.

25. Використовуючи макроси обробки змінної кількості параметрів, скласти програму сортування чисел (використати алгоритм швидкого сортування).

26. Використовуючи директиви умовної компіляції, на вимогу автора забезпечити виключення деяких частин коду. Продемонструвати застосування умовної компіляції на прикладі будь-якої раніше складеної програми.

27. Використовуючи макроси класифікації символів ASCII, визначити алфавітні, цифрові символи, символи пунктуації та інші під час читання текстового файлу. Вивести на екран символи за порядком: цифровий,

алфавітний, символ пунктуації, цифровий, алфавітний. На випадок неоднакової кількості символів різних типів вказати кількість символів, яких немає. Всі виведення оформити у вигляді макросів.

28. Написати програму, що імітує роботу калькулятора (елементарні арифметичні функції), використовуючи макроси.

Контрольні питання

1. Дайте означення макросу.
2. Як записати макрос?
3. Як виконується макрос?
4. У чому полягає відмінність між макросом та функції?
5. Які макроси обробляють змінну кількість параметрів функції?
6. Які макроси визначають значення символів?
7. Що таке «наперед означені макроси»?

Лабораторна робота №17 «Графіка»

1. Дослідити область визначення і побудувати графік функції $y = 1 / x$.
2. Дослідити область визначення і побудувати графік функції $y = (x + 3) / (x - 2)$.
3. Дослідити область визначення і побудувати графік функції $y = 3 + 2 / x - 3 / x$.
4. Дослідити область визначення і побудувати графік функції $y = 1 / (3x^2 + 2x + 1)$.
5. Дослідити область визначення і побудувати графік функції $y = x / (x^2 + x + 1)$.
6. Дослідити область визначення і побудувати графік функції $y = x / (x^2 - 3x - 1)$.
7. Дослідити область визначення і побудувати графік функції $y = x / (x^2 + 3x + 1)$.
8. Дослідити область визначення і побудувати графік функції $y = x * \arctg (x)$.
9. Дослідити область визначення і побудувати графік функції $y = 3x^4 + 4x^3 - 12x^2 - 5$.
10. Дослідити область визначення і побудувати графік функції $y = 2^x + 5x - 3$.
11. Дослідити область визначення і побудувати графік функції $y = 0.5^x - (x-2)^2 + 1$.
12. Дослідити область визначення і побудувати графік функції $y = (x - 3) \cos (x) - 1$.
13. Дослідити область визначення і побудувати графік функції $y = \ln (x) + (x + 1)^3$.
14. Дослідити область визначення і побудувати графік функції $y = x^3 + 2x^2 + 2$.
15. Дослідити область визначення і побудувати графік функції $y = x - \sin (x) - 0.25$.
16. Дослідити область визначення і побудувати графік функції $y = \ln (\sqrt{x})$.
17. Дослідити область визначення і побудувати графік функції $y = x^3 - 3x^2 + 9x - 8$.
18. Дослідити область визначення і побудувати графік функції $y = |\sin (x)| + \cos |x|$.
19. Дослідити область визначення і побудувати графік функції $y = \sqrt{x^4 + 1}$.
20. Дослідити область визначення і побудувати графік функції $y = ((x + 2)^2)^{1/3} - ((x-2)^2)^{1/3}$.
21. Дослідити область визначення і побудувати графік функції $y = 2 \sin (x) e^{x/5}$.
22. Дослідити область визначення і побудувати графік функції $y = \sin^2 x / \sqrt{1 + 2 \cos (x)}$.
23. Дослідити область визначення і побудувати графік функції $y = - \lg (x + 2)$.
24. Дослідити область визначення і побудувати графік функції $y = 2 \arccos (x-1)$.
25. Дослідити область визначення і побудувати графік функції $y = 3 \arctg (x + 2)$.
26. Дослідити область визначення і побудувати графік функції $y = (1/3)^{x-1}$.
27. Дослідити область визначення і побудувати графік функції $y = (1/2) \cos (3x + 3\pi / 4)$.
28. Дослідити область визначення і побудувати графік функції $y = |\sin (x)| - \cos |x|$.
29. Дослідити область визначення і побудувати графік функції $y = (x + 5) / (x - 7)$.
30. Дослідити область визначення і побудувати графік функції $y = (x - 3) \sin (x) - 1$.

Вказівки до виконання роботи

При побудові графіків функцій потрібно враховувати наступне:

- 1) на початку роботи програми користувач задає інтервал (максимальне та мінімальне значення по вісі x), в межах якого має бути побудований графік. Виходячи з цього, можна визначити, як змінюється функція на

заданому інтервалі (знайти її мінімальне та максимальне значення), а потім вирахувати коефіцієнти масштабування по вісях x та y таким чином, щоб графік максимально займав площину вікна;

2) якщо в область побудови потрапляють вісі абсцис та ординат, вони мають бути побудовані та на них мають бути відмічені одиничні інтервали відповідно до масштабу. Якщо ж вони не потрапляють в область побудови – одиничні інтервали відмічаються вздовж краю вікна, в якому побудовано графік;

3) у користувача має бути можливість переміщувати графік вправо-вліво-вгору-вниз, а також наближати та віддаляти його (тобто змінювати масштаб).