**ЛАБОРАТОРНА РОБОТА № 23**

**Варіант 2**

Робота з рядковими та символьними змінними

***Мета*** : Формування навиків роботи із рядковими змінними та розробки алгоритмів їх обробки. Отримання практичних навиків при передачі рядків у функцію

**Хід роботи:**

**Завдання 1**: Написати програму та протестувати. В програмі використовується масив символів. Дано рядок. Підрахувати в рядку кількість прописних (малих) латинських букв.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include<string.h>

#include<ctype.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char a[100];

int b = 0;

gets\_s(a);

for (int i = 0; i < strlen(a); i++)

{

if (islower(a[i]))

b++;

}

printf("Кількість малих літер --- %d\n", b);

return 0;

}



**Завдання 2**: Написати програму та протестувати. В програмі використовується масив символів. Речення складається із деякої кількості слів, розділених одним або декількома пробілами. Видалити із речення лишні пробіли: там де їх більше одного.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include<string.h>

#include<ctype.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char a[100];

gets\_s(a);

int t = 0, b = 0;;

for (int i = 0; i < strlen(a); i++)

{

if (isspace(a[i]))

b++;

else b = 0;

if (b > 1) {

t++;

for (int j = i; j < strlen(a); j++)

a[j] = a[j + 1];

i = 0;

}

}

puts(a);

return 0;

}



**Завдання 3**: Написати програму та протестувати. В програмі використовується масив символів. Речення складається із деякої кількості слів, розділених пробілами та певною кількістю ком. Знайти в реченні введене із клавіатури слово та замінити його іншим, заданим із клавіатури словом. Результат вивести на екран.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include<string.h>

#include<ctype.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int k = 0;

char\* sl, \*sl1, c[100], b[100], c1[100], b1[100], str2[100] = "";

char str[] = "There were mashed potatoes, boiled eggs, fish and chips, and spaghetti";

puts(str);

printf("\nВведіть слово, яке хочете замінити --- \n");

gets\_s(b);

printf("Введіть нове слово --- \n");

gets\_s(c);

strcpy(b1, b);

strcpy(c1, c);

strcat(b1, ",");

strcat(c1, ",");

sl = strtok(str, " ");

while (sl != NULL) {

if (strcmp(sl, b) == NULL)

strcat(str2, c);

else {

if (strcmp(sl, b1) == NULL)

strcat(str2, c1);

else

strcat(str2, sl);

}

strcat(str2, " ");

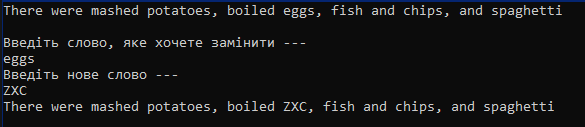
sl = strtok(NULL, " ");

}

puts(str2);

return 0;

}



**Завдання 4**: З клавіатури вводиться текстовий рядок. Розробити програму, що використовує покажчики:

а) перевіряє, чи співпадає кількість відкритих і закритих дужок у введеному рядку (перевірити для круглих та квадратних дужок);

б) виводить на екран найдовше слово;

в) видаляє всі слова, що складаються тільки з латинських літер.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include<string.h>

#include<ctype.h>

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int k1 = 0, k2 = 0;

char str[1000], str1[1000], str2[1000] = "";

gets\_s(str);

strcpy(str1, str);

puts("Завдання a)Круглі");//1

for (int i = 0; i < strlen(str); i++)

{

if (str[i] == '(')

k1++;

if (str[i] == ')')

k2++;

}

if (k1 == k2)

puts("Кількість відкритих і закритих дужок співпадає");

if (k1 > k2)

puts("Відкритих дужок більше, ніж закритих");

if (k1 < k2)

puts("Закритих дужок більше, ніж відкритих");

k1 = 0;

k2 = 0;

puts("\nЗавдання a)Квадратні");

for (int i = 0; i < strlen(str); i++)

{

if (str[i] == '[')

k1++;

if (str[i] == ']')

k2++;

}

if (k1 == k2)

puts("Кількість відкритих і закритих дужок співпадає");

if (k1 > k2)

puts("Відкритих дужок більше, ніж закритих");

if (k1 < k2)

puts("Закритих дужок більше, ніж відкритих");

puts("\nЗавдання б)");//2

char\* sl, \*maxs = (char\*)malloc(sizeof(int) \* 1), count, max = 0;

sl = strtok(str, " ,.-");

while (sl != NULL) {

if (strlen(sl) > max) {

max = strlen(sl);

strcpy(maxs, sl);

}

sl = strtok(NULL, " ,.-");

}

puts(maxs);

puts("\nЗавдання в)");//3

sl = strtok(str1, " ,.-");

while (sl != NULL) {

count = 0;

for (int i = 0; i < strlen(sl); i++) {

if ((sl[i] >= 'a' && sl[i] <= 'z') || (sl[i] >= 'A' && sl[i] <= 'Z'))

count++;

}

if (count != strlen(sl)) {

strcat(str2, sl);

strcat(str2, " ");

}

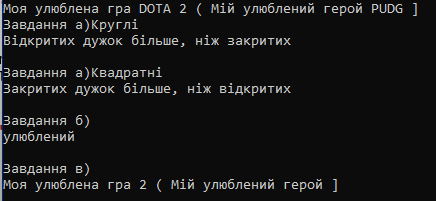
sl = strtok(NULL, " ,.-");

}

puts(str2);

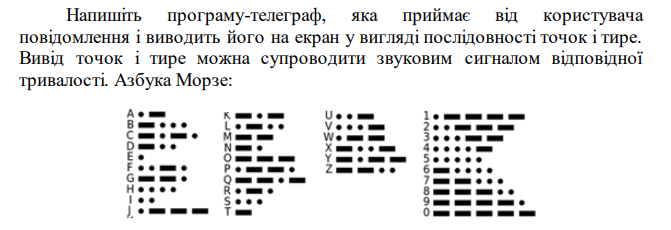
return 0;

}



**Самостійна робота:**

**Завдання 1**:



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <windows.h>

#include<stdio.h>

#include<math.h>

#include <conio.h>

#include <stdlib.h>

#include<string.h>

#include<ctype.h>

#include <cstdio>

#define N 1000

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s[N], a[N] = "";

printf("Введіть: ");

gets\_s(s);

int n = N;

for (int i = 0; i < strlen(s); i++) {

if (isupper(s[i]))

s[i] = tolower(s[i]);

}

for (int i = 0; i < strlen(s); i++) {

switch (s[i]) {

case 'a':

strcat(a, ". - ");

break;

case 'b':

strcat(a, "- . . . ");

break;

case 'c':

strcat(a, "- . - . ");

break;

case 'd':

strcat(a, "- . . ");

break;

case 'e':

strcat(a, ". ");

break;

case 'f':

strcat(a, ". . - . ");

break;

case 'g':

strcat(a, "- - . ");

break;

case 'h':

strcat(a, ". . . . ");

break;

case 'i':

strcat(a, ". . ");

break;

case 'j':

strcat(a, ". - - - ");

break;

case 'k':

strcat(a, "- . -");

break;

case 'l':

strcat(a, ". - . . ");

break;

case 'm':

strcat(a, "- - ");

break;

case 'n':

strcat(a, "- . ");

break;

case 'o':

strcat(a, "- - - ");

break;

case 'p':

strcat(a, ". - - .");

break;

case 'q':

strcat(a, "- - . -");

break;

case 'r':

strcat(a, ". - . ");

break;

case 's':

strcat(a, ". . .");

break;

case 't':

strcat(a, "- ");

break;

case 'u':

strcat(a, ". . - ");

break;

case 'v':

strcat(a, ". . . - ");

break;

case 'w':

strcat(a, ". - - ");

break;

case 'x':

strcat(a, "- . . - ");

break;

case 'y':

strcat(a, "- . - - ");

break;

case 'z':

strcat(a, "- - . . ");

break;

case '0':

strcat(a, "- - - - - ");

break;

case '1':

strcat(a, ". - - - - ");

break;

case '2':

strcat(a, ". . - - - ");

break;

case '3':

strcat(a, ". . . - - ");

break;

case '4':

strcat(a, ". . . . - ");

break;

case '5':

strcat(a, ". . . . . ");

break;

case '6':

strcat(a, "- . . . . ");

break;

case '7':

strcat(a, "- - . . . ");

break;

case '8':

strcat(a, "- - - . . ");

break;

case '9':

strcat(a, "- - - - . ");

break;

case ' ':

strcat(a, " / ");

break;

default:

strcat(a, "");

break;

}

}

puts(a);

for (int i = 0; i < strlen(a); i++) {

if (a[i] == '.') {

Beep(659.26, 200);

}

if (a[i] == '-') {

Beep(497, 500);

}

if (a[i] == ' ' || a[i] == '/') {

Sleep(200);

}

}

}



***Висновки*** я сформував навички роботи із рядковими змінними та розробки алгоритмів їх обробки. Отримання практичних навиків при передачі рядків у функцію