**ЛАБОРАТОРНА РОБОТА №1**

**Робота за базовими типами даних**

***Мета:*** отримати практичні навички з роботи з базовими типами даних (простими та зіставними)

**Зміст роботи**

Завдання 1. Записав та заповнив структуру даних зберігання дати та часу (включно зі секундами) у найкомпактнішому вигляді

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include <time.h>

#include <string.h>

#include <conio.h>

struct Date {

unsigned char Year : 7;

unsigned char Day : 5;

unsigned char WeekDay : 3;

unsigned char Month : 4;

unsigned char Hour : 5;

unsigned char Minutes : 6;

unsigned char Seconds : 6;

};

enum Week\_Days

{

Monday,

Tuesday,

Wednesday,

Thursday,

Friday,

Saturday,

Sunday

};

int Day\_Flag(int day, int month, int year) {

if (day > 0) {

if (month == 2) {

if (year % 4 == 0) {

if (day <= 29) {

return false;

}

else return true;

}

else

if (day <= 28) {

return false;

}

else return true;

}

else if ((month < 8 && month % 2 == 0) || (month > 8 && month % 2 != 0)) {

if (day <= 30) {

return false;

}

}

else if (day <= 31) {

return false;

}

else return true;

}

else return true;

}

int main(int argc, char\*\* argv[])

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Date date;

int n;

do {

printf("Уведіть рік (до 20-ти - 21-ре ст (2015 = 15), до 99 - 19 (1921 = 21), літери зупинять програму): "); scanf\_s("%d", &n);

} while (n > 99 || n < 0);

date.Year = n;

do {

printf("Уведіть місяць: "); scanf\_s("%d", &n);

} while (n > 12 || n < 1);

date.Month = n;

int year;

time\_t t;

tm\* now;

time(&t);

now = localtime(&t);

int local\_year = now->tm\_year - 100;

if (date.Year <= local\_year) {

year = date.Year + 2000;

}

else {

year = date.Year + 1900;

}

do {

printf("Уведіть день згідно місяця та року: "); scanf\_s("%d", &n);

} while (Day\_Flag(n, date.Month, date.Year));

date.Day = n;

do {

printf("Уведіть день тижня (на дійсність не зважайте): "); scanf\_s("%d", &n);

n--;

} while (n > 6 || n < 0);

date.WeekDay = n;

do {

printf("Уведіть годину: "); scanf\_s("%d", &n);

} while (n > 23 || n < 0);

date.Hour = n;

do {

printf("Уведіть хвилину: "); scanf\_s("%d", &n);

} while (n > 59 || n < 0);

date.Minutes = n;

do {

printf("Уведіть секунду: "); scanf\_s("%d", &n);

} while (n > 59 || n < 0);

date.Seconds = n;

char day[11];

switch (date.WeekDay)

{

case Monday:

strcpy(day, "понеділок");

break;

case Tuesday:

strcpy(day, "вівторок");

break;

case Wednesday:

strcpy(day, "середа");

break;

case Thursday:

strcpy(day, "четвер");

break;

case Friday:

strcpy(day, "п\'ятниця");

break;

case Saturday:

strcpy(day, "субота");

break;

case Sunday:

strcpy(day, "неділя");

break;

default:

break;

}

printf("Ваша дата:\n%04d.%02d.%02d %02d:%02d:%02d,\n%s\n\n", year, date.Month, date.Day, date.Hour, date.Minutes, date.Seconds, day);

\_getch();

}

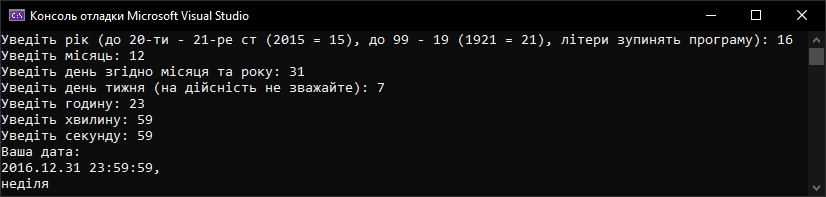


Рис.1.1 – перше завдання

Завдання 2. Реалізував введення цілочисельного значення типу signed short, визначив знак, використовуючи:

1. Логічні операції
2. Визначення структури даних

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include <conio.h>

union sShort

{

signed short num;

struct Bites {

unsigned short value : 15;

unsigned short sign : 1;

} bites;

} sshort;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Введіть число: "); scanf\_s("%d", &(sshort.num));

printf("Структура даних: ");

if (sshort.bites.sign) {

printf("число від'ємне\n");

}

else {

printf("число додатнє\n");

}

printf("Логічна операція: ");

if (sshort.num < 0) {

printf("число від'ємне\n");

}

else {

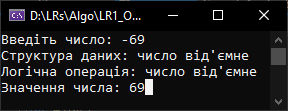
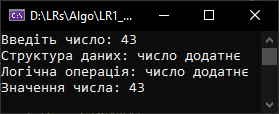
printf("число додатнє\n");

}

printf("Значення числа: %d", abs(sshort.num));

\_getch();

}

а – від'ємне число б – додатнє число

Рис.1.2 – друге завдання

Завдання 3. Зробив операції для змінних типу signed char, виповнив перевірку та пояснив випадки за наступними даними:

1. 5+127
2. 2-3
3. -120-34
4. (unsigned char)-5
5. 56 &38
6. 56 | 38

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include <conio.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

signed char n;

n = 5 + 127;

printf("5 + 127 = %d\n", n);

n = 2 - 3;

printf("2 - 3 = %d\n", n);

n = -120 - 34;

printf("-120 - 34 = %d\n", n);

n = unsigned char(-5);

printf("unsigned char(-5) = %d\n", n);

n = 56 & 38;

printf("56 & 38 = %d\n", n);

n = 56 | 38;

printf("56 | 38 = %d\n", n);

printf("\nПояснення: у знакових типах даних за виходу за межі значення змінює знак через бітову складову, де перший біт - знаковий.\nЗа виходу за значення він змінюється, як і число\n");

printf\_s("\nЗначення за int:\n");

int b;

b = 5 + 127;

printf("5 + 127 = %d\n", b);

b = 2 - 3;

printf("2 - 3 = %d\n", b);

b = -120 - 34;

printf("-120 - 34 = %d\n", b);

b = unsigned char(-5);

printf("unsigned char(-5) = %d\n", b);

b = 56 & 38;

printf("56 & 38 = %d\n", b);

b = 56 | 38;

printf("56 | 38 = %d\n", b);

\_getch();

}

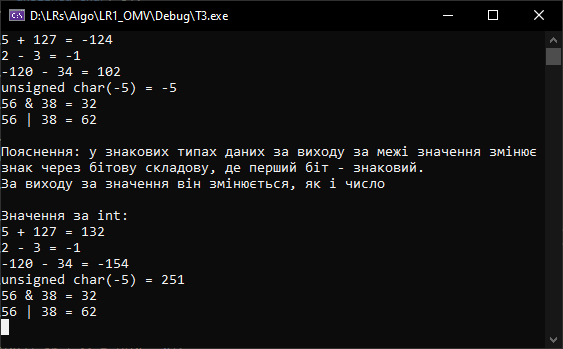


Рис.1.3 – третє завдання

Завдання 4: Створив структуру даних для зберігання дійсного числа типу float у найкомпактнішому вигляді. Вивів значення побітово, побайтово, вказуючи знак, об’єм структури тощо.

Лістинг програми:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <Windows.h>

#include <stdio.h>

#include <math.h>

#include <conio.h>

union Float {

float x;

struct Bites {

unsigned int bite32 : 1;

unsigned int bite31 : 1;

unsigned int bite30 : 1;

unsigned int bite29 : 1;

unsigned int bite28 : 1;

unsigned int bite27 : 1;

unsigned int bite26 : 1;

unsigned int bite25 : 1;

unsigned int bite24 : 1;

unsigned int bite23 : 1;

unsigned int bite22 : 1;

unsigned int bite21 : 1;

unsigned int bite20 : 1;

unsigned int bite19 : 1;

unsigned int bite18 : 1;

unsigned int bite16 : 1;

unsigned int bite17 : 1;

unsigned int bite15 : 1;

unsigned int bite14 : 1;

unsigned int bite13 : 1;

unsigned int bite12 : 1;

unsigned int bite11 : 1;

unsigned int bite10 : 1;

unsigned int bite9 : 1;

unsigned int bite8 : 1;

unsigned int bite7 : 1;

unsigned int bite6 : 1;

unsigned int bite5 : 1;

unsigned int bite4 : 1;

unsigned int bite3 : 1;

unsigned int bite2 : 1;

unsigned int bite1 : 1;

} bi;

struct Bytes {

unsigned int byte4 : 8;

unsigned int byte3 : 8;

unsigned int byte2 : 8;

unsigned int byte1 : 8;

} by;

}fl;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

printf("Введіть число: "); scanf\_s("%f", &fl.x);

printf\_s("\nБіти: %d%d%d%d %d%d%d%d %d%d%d%d %d%d%d%d %d%d%d%d %d%d%d%d %d%d%d%d %d%d%d%d\n",

fl.bi.bite1, fl.bi.bite2, fl.bi.bite3, fl.bi.bite4, fl.bi.bite5, fl.bi.bite6, fl.bi.bite7, fl.bi.bite8, fl.bi.bite9,

fl.bi.bite10, fl.bi.bite11, fl.bi.bite12, fl.bi.bite13, fl.bi.bite14, fl.bi.bite15, fl.bi.bite16, fl.bi.bite17,

fl.bi.bite18, fl.bi.bite19, fl.bi.bite20, fl.bi.bite21, fl.bi.bite22, fl.bi.bite23, fl.bi.bite24, fl.bi.bite25,

fl.bi.bite26, fl.bi.bite27, fl.bi.bite28, fl.bi.bite29, fl.bi.bite30, fl.bi.bite31, fl.bi.bite32);

printf\_s("\nБайти: %02x %02x %02x %02x\n", fl.by.byte1, fl.by.byte2, fl.by.byte3, fl.by.byte4);

printf\_s("\nЗнак: %d, %c", fl.bi.bite1, fl.bi.bite1 ? '-' : '+');

printf("\nОб\'єм пам\'яті: %d\n", sizeof(fl));

\_getch();

}

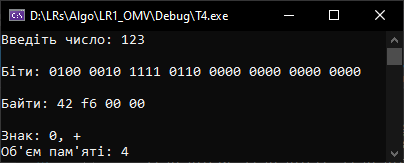
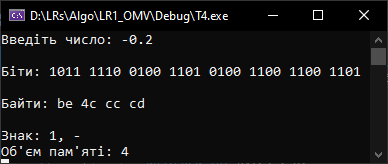
   
а – додатнє число б – від’ємне число

Рис.1.4 – четверте завдання

***Висновок:*** було отримано практичні навички роботи з різними базовими типами даних