Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут телекомунікацій, радіоелектроніки та електронної техніки



Звіт до лабораторної роботи №11

з дисципліни «Програмування, частина 1»

Виконав:

ст. гр. АП-11

Білоніжко Віталій

Викладач: Чайковський І. Б

Львів-2022

Мета роботи: Дослідження властивостей, аргументів і зовнішніх змінних

функцій мови програмування С. Оволодіння механізмом визначення, опису і виклику функцій та створення рекурсивних функцій

1.

// Скласти програму, що складається з функції main та двох функцій. Перша функція ініціалізує масив значеннями арифметичної прогресії. Друга функція обчислює суму елементів масиву. Масив та кількість його елементів оголосити глобально, щоб обидві функції мали доступ до масиву без передачі через формальні параметри.

#include <stdio.h>

#include <stdlib.h>

int init(int \*a, int n);

int sum(int \*a, int n);

int main()

{

int n;

printf("Enter n: ");

scanf("%d", &n);

int \*a = (int \*)malloc(n \* sizeof(int));

init(a, n);

printf("sum = %d\n", sum(a, n));

free(a);

return 0;

}

int init(int \*a, int n)

{

for (int i = 0; i < n; i++)

{

a[i] = i;

}

}

int sum(int \*a, int n)

{

int s = 0;

for (int i = 0; i < n; i++)

{

s += a[i];

}

return s;

}

2.

// Програма main повинна надрукувати суму прогресії

// Використавши функцію скласти програму виведення на екран залізничного квитка для проїзду до пункту призначення z зони за ціною m гривень за проїзд однієї зони. Номер зони z ввести в режимі діалогу.

#include <stdio.h>

int main()

{

int z, m;

printf("Enter z: ");

scanf("%d", &z);

printf("Enter m: ");

scanf("%d", &m);

printf("Ticket to zone %d costs %d\n", z, z \* m);

return 0;

}

3.

// В покроковому режимі дослідити роботу програми обчислення факторіалу за допомогою рекурсивного виклику функції:

// int fact (int n) { return ((n > 1) ? n\*fact(n-1) : 1 ); }

#include <stdio.h>

#include <stdlib.h>

// int fact (int n) { return ((n > 1) ? n\*fact(n-1) : 1 ); }

int fact(int n)

{

// Перевірка на вхідні дані чи більше 1 якщо ні то повертаємо 1

if (n > 1)

// Якщо більше 1 то викликаємо функцію з параметром n-1 і множимо на n це все відбувається поки n не стане 1.

// Результат виклику функції записуємо в return.

// Рекурсивний виклик функції.

return n \* fact(n - 1);

else

return 1;

}

int main()

{

int n;

printf("Enter n: ");

scanf("%d", &n);

int \*a = (int \*)malloc(n \* sizeof(int));

// factorial use for without recursion

int f = 1;

for (int i = 1; i <= n; i++)

{

f \*= i;

}

printf("factorial = %d\n", f);

printf("Test fact(%d) = %d\n", n, fact(n));

return 0;

}

4.

// Здійснити модифікацію функції fact таким чином, щоб вона повертала значення типу double.

#include <stdio.h>

double fact(int n)

{

if (n > 1)

return n \* fact(n - 1);

else

return 1;

}

int main()

{

int n;

printf("Enter n: ");

scanf("%d", &n);

printf("Test fact(%d) = %f\n", n, fact(n));

return 0;

}

5.

// Використавши функцію скласти програму заповнення екрану символами, що утворюють прямокутний трикутник, катети якого задати у вигляді діалогу.

#include <stdio.h>

void print\_triangle(int a, int b)

{

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

{

for (int j = 0; j < b; j++)

{

if (i == 0 || i == a - 1 || j == 0 || j == b - 1)

printf("\*");

else

printf(" ");

}

printf("\n");

}

}

int main()

{

int a, b;

printf("Enter a: ");

scanf("%d", &a);

printf("Enter b: ");

scanf("%d", &b);

print\_triangle(a, b);

return 0;

}

Висновок: на даній лабораторній роботі я навчився досліджувати властивості, аргументів і зовнішніх змінних функцій мови програмування С. Оволодів механізмом визначення, опису і виклику функцій та створення рекурсивних функцій