**Указатели на функции**

**Цель работы**: разработать и правильно поставить задачи на примерах из лекции по указателям на функции.

**Задача 1**

**Постановка задачи**: привести несколько возможных вариантов указателей на функции и описать их.

**Программный код**:

int (\*fn1)(int);

char (\*fn2)(int, char);

double (\*fn3)(int, char, double);

void (\*p1)(int);

void (\*p2)(int, char);

void (\*p3)(int, char, double);

int (\*v1)(void);

char (\*v2)(void);

double (\*v3)(void);

// Объявление указателя fn1 на функцию типа integer, которая принимает один аргумент типа integer.

int (\*fn1)(int);

// Объявление указателя fn2 на функцию типа char, которая принимает два аргумента типов integer и char соответственно.

char (\*fn2)(int, char);

// Объявление указателя fn3 на функцию типа double, которая принимает три аргумента типов integer, char, double соответственно.

double (\*fn3)(int, char, double);

// Объявление указателя p1 на процедуру, которая принимает один аргумент типа integer.

void (\*p1)(int);

// Объявление указателя p2 на процедуру, которая принимает два аргумента типов integer и char соответственно.

void (\*p2)(int, char);

// Объявление указателя p3 на процедуру, которая принимает три аргумента типов integer, char, double соответственно.

void (\*p3)(int, char, double);

// Объявление указателя v1 на функцию типа integer, которая не принимает аргументов.

int (\*v1)(void);

// Объявление указателя v2 на функцию типа char, которая не принимает аргументов.

char (\*v2)(void);

// Объявление указателя v3 на функцию типа double, которая не принимает аргументов.

double (\*v3)(void);

**Задача 2**

**Постановка задачи**: реализовать программу, которая выполняет простые арифметические операции над двумя целочисленными аргументами, такие как: сложение(+), вычитание(-),умножение(\*), деление(/). При этом для каждой операции организовать функцию, вызов которых будет производиться через указатели, также создать процедуру, которая выводит общую информацию про функционал программы, и вызов которой будет осуществляться через указатель.

**Программный код**:

#include<stdio.h>

#include<stdlib.h>

#include <locale.h>

// Объявление прототипа процедуры, выводящую общую информацию

void info (void);

// Объявление прототипов функций типа integer, принимающие два аргумента типа integer

int fn1(int a, int b);

int fn2(int a, int b);

int fn3(int a, int b);

int fn4(int a, int b);

int main() {

setlocale(LC\_ALL, "Rus");

void (\*p)(void);// Объявление указателя на процедуру без аргументов.

int (\*q)(int, int);// Объявление указателя на функциу типа integer с двумя агрументами типа integer.

//Применение указателей:

p = &info; // Значение указателя

p();// Вызов процедуры с помощью указателя

q = &fn1; // Значение указателя

printf("Операции над числами 10 10:\n");

printf("Сложение: %i\n", q(10, 10));// Вызов функции с помощью указателя

q = &fn2; // Значение указателя

printf("Вычитание: %i\n", q(10, 10));// Вызов функции с помощью указателя

q = fn3; // Значение указателя

printf("Умножение: %i\n", q(10, 10));// Вызов функции с помощью указателя

q = fn4; // Значение указателя

printf("Деление: %i\n", q(10, 10));// Вызов функции с помощью указателя

system("pause");

return 0;

}

//Объявление тела прототипа

void info(void) {

printf("В данной программе можно использовать оперции:\nсложения, вычитания, умножения, деления.\n\n");

}

//Объявление тел функций

int fn1(int a, int b){

return a + b;

}

int fn2(int a, int b){

return a - b;

}

int fn3(int a, int b){

return a \* b;

}

int fn4(int a, int b) {

return a / b;

}

**Задача 3**

**Постановка задачи**: организовать программу, запрашивающую число N, до которого требуется вывести на экран числа, после чего посчитать их среднее арифметическое значение, а также суммировать весь ряд чисел.

**Программный код**:

#include<stdio.h>

#include<stdlib.h>

#include <locale.h>

int N = 0; // Глобальная переменная

// Объявление прототипа процедур и функции:

void inputN(void);

void show(int n);

float average(int n);

int main() {

setlocale(LC\_ALL, "Rus");

void (\*p)();// Объявление указателя на процедуру

float (\*q)();// Объявление указателя на функцию, которая возвращает тип float

p = &inputN; //Значение указателя

p();// Вызов процедуры

p = show; // Присваивание имени функции указателю

p(N); // Вызов функции

q = &average; // Значение указателя

printf("Среднее арифметическое ряда чисел до %i = %1.3g\n", N, q(N)); // Вызов функции с помощью указателя

system("pause");

return 0;

}

//Описание тела прототипа процедур и функции:

//Процедура задает глобальную переменную N

void inputN(void) {

printf("Введите число N: ");

scanf("%i", &N);

}

//Процедура выводит ряд чисел до n

void show(int n) {

printf("Ряд: ");

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

printf("%i ", i);

}

printf("\n");

}

// Функция возвращает тип float и считает среднее арфиметическое значение для ряда n

float average(int n) {

float sr = 0;

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

sr += i;

}

return (float)(sr / n);

}

**Задача 4**

**Постановка задачи**:

Требуется вывести на экран ряд чисел, не превышающих число N, а после этого вывести на экран этот же ряд, каждый элемент которого умножен на число X.

**Программный код**:

#include<stdio.h>

#include<stdlib.h>

#include <locale.h>

//Глобальные переменные

int N = 0;

int X = 0;

// Объявление прототипа процедур и функции:

void inputN(void);

void show(int n);

void multRow(int (\*t)(int, int),int n);

int mult(int a, int x);

int main() {

setlocale(LC\_ALL, "Rus");

void (\*p)();// Объявление указателя на процедуру

int (\*q)(int, int);// Объявление указателя на функцию, которая возвращает тип integer

p = &inputN; //Значение указателя

p();// Вызов процедуры

p = show; // Присваивание имени функции указателю

p(N); // Вызов функции

q = &multRow; // Значение указателя

q(mult, N); // Вызов функции с помощью указателя

system("pause");

return 0;

}

//Описание тела прототипа процедур и функции:

//Процедура задает глобальные переменные N и X

void inputN(void) {

printf("Введите ряд N и множитель X: ");

scanf("%i%i", &N, &X);

}

//Процедура выводит ряд чисел до n

void show(int n) {

printf("Ряд: ");

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

printf("%i ", i);

}

printf("\n");

}

// Процедура перемножает каждый член ряда на X и выводит на экран

void multRow(int (\*t)(int, int), int n) {

printf("Ряд умноженный на %i: ", X);

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

printf("%i ", t(i,X));

}

printf("\n");

}

// Функция имеет два аргумента типа integer, она возвращает тип integer и произведение входящих в неё аргументов.

int mult(int a, int x) {

return a \* x;

}