## Лабораторная работа 9

## Задание 1

Объявите две переменные целого типа. На каждую из них

ссылается указатель. На первую ссылается указатель p\_1, а на

вторую указатель p\_2. Кроме того, объявлена переменная

типа double и указатель на неѐ p\_dbl.

Используя указатели, подсчитать частное целых переменных

(первую делим на вторую) и сохранить это значение в переменную,

на которую ссылается p\_dbl.

## Код программы

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

int a=2,b=1, \*p\_1=&a,\*p\_2=&b;

double x, \*p\_dbl=&x;

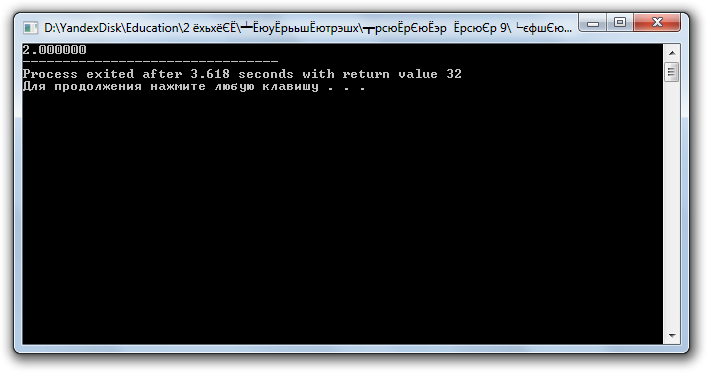
\*p\_dbl=\*p\_1/(\*p\_2);

printf("%f",\*p\_dbl);

getch();

}

## Результат проделанной работы



## Задание 2

Что выполняет приведенная программа? Найдите ошибку в

программе и объясните ее причину. Исправьте программу так,

чтобы она работала корректно.

#include <stdio.h>

void swap(int a, int b) {

int temp = a;

a = b;

b = temp;

}

int main(){

int x = 3,y = 5;

printf("x=%d \t y=%d \n",x,y);

swap(x,y);

printf("x=%d \t y=%d \n",x,y);

return (0);

}

## Код программы

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main(){

int x = 3,y = 5;

printf("x=%d \t y=%d \n",x,y);

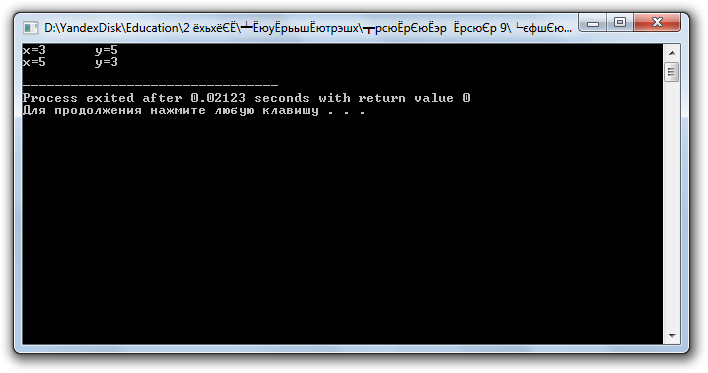
swap(&x,&y); //Изначально было

printf("x=%d \t y=%d \n",x,y);

return (0);

}

## Результат проделанной работы



## Задание 3

Что выполняется в данной программе? Дайте ответ и напишите

комментарии к программе.

## Код программы

#include <conio.h>

#include <conio.h>

#include <stdio.h>

void main() {

int length;

char \*p1, \*p2;

char tmp;

float a = 5.0f;

float b = 3.0f;

printf("a = %.3f\n", a); //Вывод переменной с 3 знаками после запятой

printf("b = %.3f\n", b); //Вывод переменной с 3 знаками после запятой

p1 = (char\*) &a; //Сохранение адреса переменной в типе char

p2 = (char\*) &b; //Сохранение адреса переменной в типе char

length = sizeof(float); //Макс. значение float

while (length--) { //Цикл, параметром которого являеся length. Условием выхода из цикла является length равный 0

tmp = \*p1;

\*p1 = \*p2;

\*p2 = tmp; //Эта и две строки выше выполняют "обмен" значений переменных p1 и p2

p1++; //Переход на следующее значение адреса

p2++; //Переход на следующее значение адреса

}

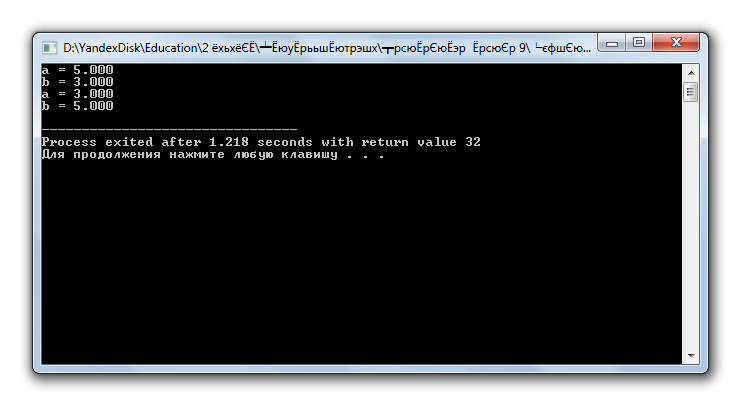
printf("a = %.3f\n", a); // Вывод переменной с 3 знаками после запятой

printf("b = %.3f\n", b); // Вывод переменной с 3 знаками после запятой

getch();

}

## Результат проделанной работы



## Задание 4

Допустимо ли в Си? Если "да" - опишите семантику каждого

правильного действия (не принимая во внимание ошибочные); если

"нет" - объясните почему.

a).

int i, ∗ p, j, ∗q;

p = &i; q = &p;

j = ∗p = 1; q = p-1; \*p += 1;

i = ∗++q + ∗p; q -= 1; i = ∗q ++ + ∗q;

printf("i=%d, j=%d, ∗p=%d, ∗q=%d \n", i, j, ∗p, ∗q);

b).

int x = 1, y;

char c = ‘a’; int ∗pi, ∗qi; char ∗pc;

pi = &x; ∗pi = 3; y = ∗pi; ∗pi = c; qi = pi;

pc = qi; ∗qi+=1; pi++; ∗(- - pi) = 5; y = ∗qi+1;

pc = &c; ++\*pc; (∗pc)++; ∗pc++; ∗pc+=1;

x = (int)pi; pi=(int∗)pc; pi=(int∗)x; x = 1+ ∗pi; pc=(char∗)pi;

c = ∗pc; pc = &y; x = qi – pi; qi = 0; qi+=pi;

y = &pi; y = (int)&pi; pi = pi +5; ∗(pi+1)=0; pi=&(x+0);

## (а)

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

int i, \*p, j, \*q;

p = &i; q = &p; //Указателям присваиваем адреса переменных i и p

j = \*p = 1; //Присваивание переменнной j=i=1

q = p-1; //Присваиваем узателю адрес переменной, которая стоит перед адресом переменной i

\*p += 1; //Присваиваем i значение переменной, адрес которой стоит после i

i = \*++q + \*p; //Присваиваем i значение переменной, адрес которой стоит после i+1

q -= 1; //Присваиваем q начение адреса, который находится перед q

i = \*q ++ + \*q; //Присваиваем переменной i значение (пред.адрес i+пред. адрес i) + 1

printf("i=%d, j=%d, \*p=%d, \*q=%d \n", i, j, \*p, \*q);

getch();

}

## Код программы (б)



Т.е. нельзя выполнять эти операции с данными типами данных