**Лабораторная работа № 7.**

**Указатели на простейшие типы. Операции над указателями.**

1. Постановка задачи

Заполните пустые места в программе таким образом, чтобы программа выдавала полученный результат.

Код программы

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,c,max;

int \*pa=&a, /\* Переменная pa содержит адрес переменной a \*/

\*pb=&b, /\* Переменная pb содержит адрес переменной b \*/

\*pc=&c, /\* Переменная pc содержит адрес переменной c \*/

\*min;

printf("Введите значения a,b,c: ");

scanf("%d %d %d",&a,&b,&c);

printf("Посмотрим, по каким адресам лежат эти значения:\n");

printf(" в ячейке с адресом %p лежит %d\n",pa,a);

printf(" в ячейке с адресом %p лежит %d\n",pb,b);

printf(" в ячейке с адресом %p лежит %d\n",pc,c);

printf("Найдем большее из чисел a, b, c и положим\n");

printf("в ячейку с меньшим адресом.\n");

/\* Поиск наименьшего адреса \*/

if (pa<=pb)

min=pa;

else min=pb;

if (min>=pc)

min=pc;

/\*Допишите блок кода, чтобы был полностью реализован поиск наименьшего

адреса (не рассмотрен еще один адрес при сравнении)\*/

/\* Поиск наибольшего значения \*/

if (\*pa<=\*pb)

max=\*pb; // Объясните, почему перед ―max‖ не стоит ―\*‖

else max=\*pa;

if (max<=\*pc)

max=\*pc;

/\* Требуемое размещение \*/

/\*Положите значение наибольшего числа в ячейку с меньшим адресом

(Операция включает в себя два операнда и знак равенства между операндами) \*/

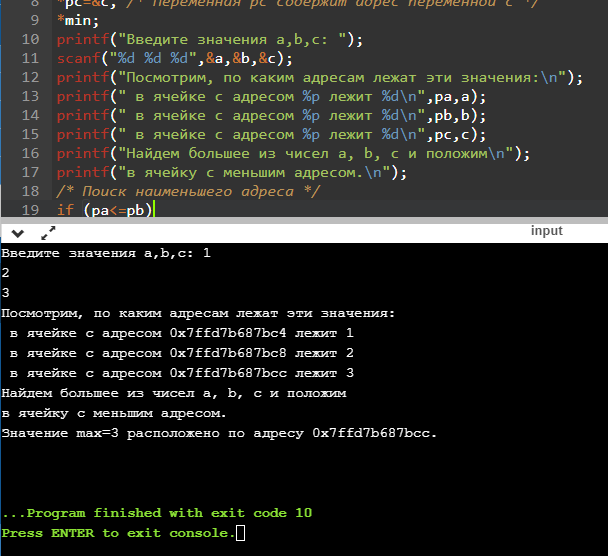
printf("Значение max=%d расположено по адресу %p.\n\n",max,pc);

getch();

}

Результаты выполненной работы

max=\*pb; // Объясните, почему перед ―max‖ не стоит ―\*‖ - адресу pb вводим значение max, не ставим перед \* max, тк это обычная переменная, а pb - указатель

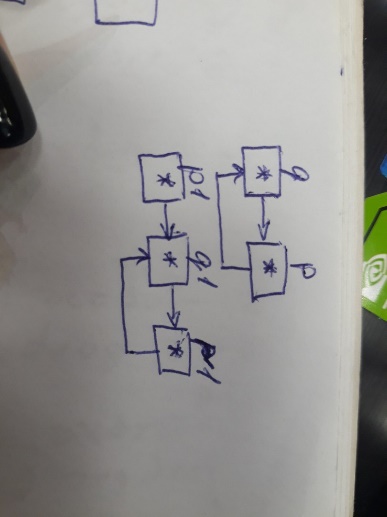


2. Постановка задачи

Графически представьте взаимосвязь указателей и их значений.

Код программы

Результаты выполненной работы



3. Постановка задачи

Найдите ошибку в использовании функции sizeof.

Код программы

#include<stdio.h>

#include<conio.h>

void main()

{

char c;

int i;

/\* ————————————————- \*/

printf("Размер типа char = %2d\n", sizeof (char));

printf("Размер типа int = %2d\n", sizeof (int));

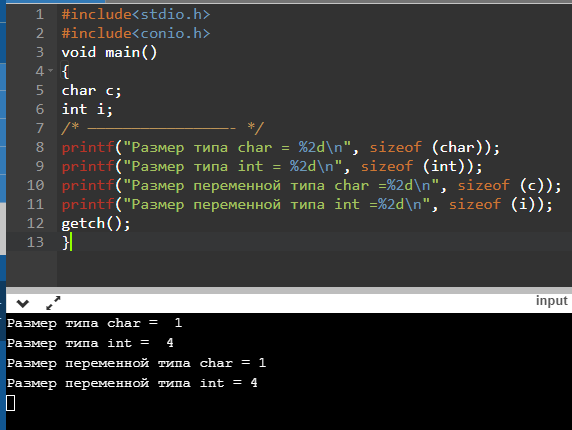
printf("Размер переменной типа char =%2d\n", sizeof (c));

printf("Размер переменной типа int =%2d\n", sizeof (i));

getch();

}

Результаты выполненной работы



4. Постановка задачи

Ознакомьтесь с программой, заполните пустые места в программе. Запустите код и сделайте скриншот результата. Ответьте на комментарии к программе.

Код программы

/\* Демонстрация разыменования указателя, который не \*/

/\* был инициализирован. Это может привести: \*/

/\* (1) к фатальной ошибке во время выполнения прог- \*/

/\* раммы; \*/

/\* (2) случайному искажению данных, в результате \*/

/\* чего программа завершается с неверным результатом \*/

/\* ————————————————- \*/

#include<stdio.h>

#include<conio.h>

void main()

{

int dec=10,

\*ptr=&dec, /\* Указатель ptr инициализирован \*/

\*ptr1; /\* Указатель ptr1 не инициализирован \*/

/\* ——————————————— \*/

printf("Указатель инициализирован:\n");

printf(" По адресу (%p) лежит переменная = %d\n",ptr,dec);

printf(" &\*ptr=%p\n \*&ptr=%p\n\n",&\*ptr,\*&ptr); // Что происходит при использовании следующих операций &\*ptr,\*&ptr

/\* —————————————– \*/

printf("Указатель не инициализирован:\n");

printf("По адресу (%p) лежит переменная = %d\n",ptr1, &ptr1 );

printf(" &\*ptr1=%p\n \*&ptr1=%p\n\n",&\*ptr1,\*&ptr1);

/\* —————————————————– \*/

printf("Инициализируем неинициализированный указатель:\n");

ptr1=&dec; // Объясните, что происходит в этой операции

printf("(%p)=%d\n\n",ptr1,dec);

getch();

}

Результаты выполненной работы

