Указатели

Задача: нужно создать структуру, описывающую информацию о сотруднике. Какого типа должна быть переменная, указывающая на начальника этого сотрудника?

Тип данных, который указывает (хранит) адрес в памяти переменной, массива или структуры (любой другой переменной) называется *указатель*.

```
struct employee
{
    char name[20];
    char fam[20];
    employee* chief;
};
```

Операции с указателями

Объявление указателя:

```
int* p;
```

Операция *взятия адреса*:

```
int a;
p = &a;
```

выполняется во время компиляции!

Операция разыменования:

```
*p = 1;
```

Адресная арифметика

Имеется указатель p, который указывает на тип размером size. При сложении указателя с числом x, новое значение указателя будет равно p+x*size

```
int a[]={0,1,2,3,4};
int b;
int* p = &a[2];
p++;
b=*p;
```

Ошибки при адресной арифметики могут привести к обращению к памяти по неверному адресу и сбою программы.

Указатели и массивы

Массив – константный (постоянный) указатель на начало массива. К нему можно применять операции над указателями (не изменяющие указатель)
Операция индексирования является операцией над указателем

```
int a[]={0, 1, 2, 3, 4};
int *p=a;
printf("%i\n", *a);
printf("%i\n", *(a+1));
printf("%i\n", p[2]);
```

Указатели и структуры

Структура – константный (постоянный) указатель на начало структуры. Над структурой нельзя применять операцию индексирования и другие операции над указателями.

Для обращения к полю a структуры s, на которую ссылается указатель p можно с помощью сокращенной формой записи оператора доступа к полю (следующие две строки эквивалентны):

```
p->a=1;
(*p).a=1;
```

Примеры работы с указателями

```
void pr_addr(int*c, int n)
{
    int i;
    printf("          ");
    for(i=0; i<n; i++)
        printf("%p ", &c[i]);
    printf("\n");
}
void dump(int*c, int n)
{
    int i;
    for(i=0; i<n; i++)
        printf("%8x ", c[i]);
    printf("\n");
}
```

Примеры работы с указателями

```
int a[3] = {1, 2, 3};
int* p1, *p2;
int x=4, y=5;

pr_addr(&y,7);
printf("              ");
dump(&y,7);

p1 = a;
printf("p1 = a;     ");
getch();
dump(&y,7);
```

Задание к лабораторной работе

1 Переделать первое задание из предыдущей лабораторной работы, используя вместо операции индексирования адресную арифметику.

Первое задание: Создать массив из 50 чисел, значения которых равны значению функции от индекса элемента. Функции взять из второго задания предыдущей лабораторной работы. Распечатать значения этого массива.

Задание к лабораторной работе

2 Для следующего фрагмента программы записать в виде таблицы значение переменных после каждого шага программы (столбцы соответствуют переменным, строки – строкам программы). Тип int и указатели занимают по 4 байта. Компилятор расположил переменные по следующим адресам: a – 100, p1 – 120, p2 – 124, s.x – 128, s.y – 132. Число N – номер варианта.