Лабораторная работа № 2

Указатели, арифметика указателей.

Задание 1.1

Постановка задачи:

Внутри функции int main(void) { ∗/... ∗/} определите указатель double ∗∗pointer = NULL;. Инициализируйте этот указатель адресом другого указателя типа double ∗, который указывает, в свою очередь, на переменную double. Используйте pointer для записи и чтения в эту перепенную значения 2

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
pointer	double ***	Указатель на указатель на переменную double.

Код программы:

```c
#include <stdio.h>
#include <stdlib.h>

int
main()
{
    double ***pointer = NULL;
    *( *( *( pointer = (double ***) malloc(sizeof(double **))) = (double **)malloc(sizeof(double *))) = (double *)malloc(sizeof(double)));
    ***pointer = 2.0;

    printf("%lf\n", ***pointer);
    free(**pointer);
    free(*pointer);
    free(pointer);

    return 0;
}
```

Результаты выполненной работы:

```
2.000000
```

Задание 1.2

Постановка задачи:

Напишите программу, которая складывает два числа с использованием указателей на эти числа.

Математическая модель:

$c = a + b$

Список идентификаторов:

Имя	**Тип**	**Назначение**
a	int	Первое число.
b	int	Второе число.
pa	int *	Указатель на первое число.
pb	int *	Указатель на второе число.
c	int	Сумма первого и второго числа.

Код программы:

```c
#include <stdio.h>

int
main()
{
    int a = 10, b = 15, c;
    int *pa =&a, *pb=&b;
    c = *pa + *pb;

    printf("%d\n", c);
    return 0;
}
```

Результаты выполненной работы:

```
25
```

Задание 1.3

Постановка задачи:

Напишите программу, которая находит максимальное число из двух чисел, используя указатели на эти числа.

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
a	int	Первое число для сравнения.
b	int	Второе число для сравнения.
pa	int *	Указатель на первое число для сравнения.
pb	int *	Указатель на второе число для сравнения.

Код программы:

```c
#include <stdio.h>

int
main()
{
    int a,b;
    scanf("%d %d", &a, &b);
    int *pa=&a, *pb=&b;

    if(*pa > *pb)
        printf("max:%d\n", *pa);
    else
        printf("max:%d\n", *pb);
    return 0;
}
```

Результаты выполненной работы:

```
10 14
max:14
```

Задание 1.4

Постановка задачи:

Напишите программу, которая создаёт одномерный динамический массив из чисел с плавающей точкой двойной точности, заполняет его значениями с клавиатуры и распечатывает все элементы этого массива, используя арифметику указателей (оператор +), а не обычный оператор доступа к элементу массива - [].

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
lenght	int	Длинна массива.
array	double *	Сам массив.
inter	double *	Указатель на итератор по которому получаем элементы массива.
end	double *	Указатель на последний элемент массива для с

Код программы:

```c
#include <stdio.h>
#include <stdlib.h>

int
main(){
    int lenght;
    printf("Enter lenght of the array:");
    scanf("%d", &lenght);

    double *array = NULL;

    array = (double *)malloc(lenght*sizeof(double));

    double *iter = array, *end = &array[lenght];

    for(iter, end; iter < end; iter++)
        scanf("%lf", iter);

    for(iter = array; iter < end; iter++)
        printf("%lf\n", *iter);

    free(array);
    return 0;
}
```

Результаты выполненной работы:

```
Enter lenght of the array:10
10
9
8
7
6
5
4
3
2
1
10.000000
9.000000
8.000000
7.000000
6.000000
5.000000
4.000000
3.000000
2.000000
1.000000
```

Задание 1.5

Постановка задачи:

Вывести элементы динамического массива целых чисел в обратном порядке, используя указатель и операцию декремента (−−).

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
a	int[10]	Массив.
iter	int *	Указатель на элемент массива.

Код программы:

```c
#include <stdio.h>
#include <stdlib.h>

int
main()
{
    int a[10] = {1,2,3,4,5,6,7,8,9,10};
    int *iter;
    for(iter = &a[9]; iter >= &a; iter--)
        printf("%d\n", *iter);
    return 0;
}
```

Результаты выполненной работы:

```
10
9
8
7
6
5
4
3
2
1
```

Задание 1.6

Постановка задачи:

Определите переменную целого типа int a = 1234567890; и выведите побайтово её содержимое на экран, используя указатель char *.

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
a	int	Переменная которую необходимо вывести.
pa	char *	Указатель на a.
i	int	Аргумент цикла, необходим чтобы не выйти за пределы a.

Код программы:

```c
#include <stdio.h>

int
main()
{
    int a = 1234567890;
    char * pa = &a;

    for(int i = 0; i < 4; i++, pa++)
        printf("%c", *pa);
    putchar('\n');
    return 0;
}
```

Результаты выполненной работы:

Задание 1.7

Постановка задачи:

Выделите память под двумерный динамический массив, используя массив указателей на строки (см. лекции). Затем освободите корректно оперативную память

Математическая модель:

-

Список идентификаторов:

Имя	**Тип**	**Назначение**
lenght	int	Длинна массива указателей.
width	int	Ширина массива указателей.
string	char **	Массив указателей.
i	int	Аргумент циклов.

Код программы:

```c
#include <stdio.h>
#include <stdlib.h>

int
main()
{
    int lenght, width;
    scanf("%d %d", &lenght, &width);

    char **string = (char **)malloc(sizeof(char *) * width);

    if(string){
        printf("Allocated string!\n");
    }else{
        printf("Malloc failed!\n");
        return 1;
    }

    for(int i = 0; i < width; i++){
        string[i] = (char *)malloc(sizeof(char) * lenght);
        if(!string){
            printf("Malloc %d failed!\n", i);
        return 1;
        }
    }

    for(int i = 0; i < width; i++)
        free(string[i]);

    free(string);
    return 0;
}
```

Результаты выполненной работы:

```
10
19
Allocated string!
```