

Лабораторная работа №2

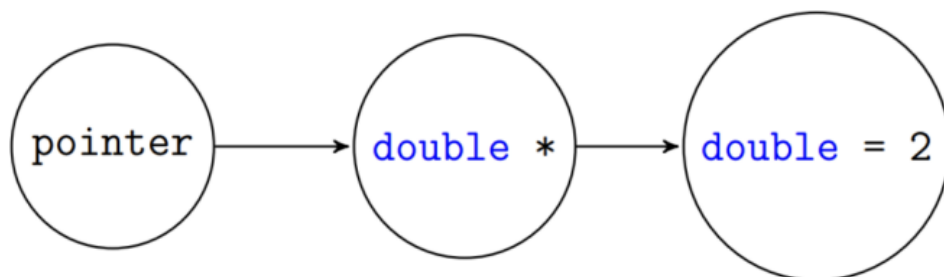
Программирование. Язык СИ.

Указатели, арифметика указателей.

Задание 1.1

Задача:

1.1: 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 переменную значения 2.



При этом выполните следующее:

- используйте функции типа `*alloc(...)` для выделения оперативной памяти под динамические объекты;
- запишите и выведите число, указанное в крайней правой окружности, на экран, используя указатель `double **pointer = NULL;`;
- используйте функцию `free(...)` для освобождения оперативной памяти, выделенной под динамические объекты.

Список идентификаторов:

Имя	Смысл	Тип
number	Переменная	double
p	Указатель	double
pointer	Указатель на указатель	double

Код программы:

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

int main(void)
{
    double number;
    double **pointer = NULL;
    double *p = (double* ) malloc(sizeof(double));
    pointer = &p;
    **pointer = 2;
    printf("%.21f", **pointer);
}
```

```
free(pointer);  
free(p);  
  
return 0;  
}
```

Результат выполнения:

```
2.00  
Process finished with exit code
```

Задание 1.2

Задача:

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

Список идентификаторов:

Имя	Смысл	Тип
a	Первое число	int
b	Второй число	int
s	Сумма чисел	int
pointer_a	Указатель на a	int
pointer_b	Указатель на b	int

Код программы:

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main(void)  
{  
    int a, b, s;  
    printf("Enter number 1: ");  
    scanf("%d", &a);  
    printf("Enter number 2: ");  
    scanf("%d", &b);  
    int *pointer_a = (int *) malloc(sizeof(int));  
    int *pointer_b = (int *) malloc(sizeof(int));  
    pointer_a = &a;  
    pointer_b = &b;  
    s = *pointer_a + *pointer_b;  
    printf("%d + %d = %d", *pointer_a, *pointer_b, s);  
    free(pointer_a);  
    free(pointer_b);  
  
    return 0;  
}
```

Результат выполнения программы:

```
Enter number 1:16
Enter number 2:34
16 + 34 = 50
Process finished with exit code
```

Задание 1.3

Задача:

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

Список идентификаторов:

Имя	Смысл	Тип
a	Первое число	int
b	Второе число	int
pointer_a	Указатель на a	int
pointer_b	Указатель на b	int

Код программы:

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

int main()
{
    int a, b;
    printf("Enter 1 number: ");
    scanf("%d", &a);
    printf("\nEnter 2 number: ");
    scanf("%d", &b);
    int *pointer_a = (int *) malloc(sizeof(int));
    int *pointer_b = (int *) malloc(sizeof(int));
    pointer_a = &a;
    pointer_b = &b;
    if (*pointer_a > *pointer_b)
    {
        printf("\nMax = %d", *pointer_a);
    }
    else if (*pointer_a < *pointer_b)
    {
        printf("\nMax = %d", *pointer_b);
    }
    else
    {
        printf("\n%d = %d", *pointer_a, *pointer_b);
    }
    free(pointer_a);
    free(pointer_b);

    return 0;
}
```

Результат выполнения программы:

```
Enter 1 number:84

Enter 2 number:63

Max = 84
Process finished with exit code
```

Задание 1.4

Задача:

1.4: 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 - [].

Список идентификаторов:

Имя	Смысл	Тип
num	Число элементов в массиве	int
i	Параметр цикла	double
mass	Массив	double

Код программы:

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

int main()
{
    double *mass;
    int num;
    printf("Enter massive size:");
    scanf("%d", &num);
    mass = (double *) malloc(num * sizeof(double));
    double *i;
    for (i = mass; i - mass < num; i++)
    {
        printf("Enter element of massive: ");
        scanf("%lf", &*i);
    }
    printf("Mass: ");
    for(i = mass; i - mass < num; i++)
    {
        printf("%.2lf ", *i);
    }
    free(mass);
    free(i);

    return 0;
}
```

Результат выполнения программы:

```
Enter massive size:6

Enter element of massive:3

Enter element of massive:14

Enter element of massive:25

Enter element of massive:84

Enter element of massive:74

Enter element of massive:6

Mass: 3.00 14.00 25.00 84.00 74.00 6.00
```

Задание 1.5

Задача:

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

Список идентификаторов:

Имя	Смысл	Тип
num	Число элементов в массиве	int
i	Параметр цикла	double
mass	Массив	double

Код программы:

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

int main()
{
    int *mass;
    int num;
    printf("Enter massive size: ");
    scanf("%d", &num);
    mass = (int *) malloc(num * sizeof(int));
    int *i;
    for (i = mass; i - mass < num; i++)
    {
        printf("\nEnter element of massive: ");
        scanf("%d", &*i);
    }
    printf("Mass: ");
    for(i = mass + num - 1; i >= mass; i--)
    {
        printf("%.d ", *i);
    }
}
```

```

    }
    free (mass);
    free (i);

    return 0;
}

```

Результат выполнения программы:

```

Enter massive size:6
Enter element of massive:12
Enter element of massive:18
Enter element of massive:91
Enter element of massive:3
Enter element of massive:75
Enter element of massive:6
Mass: 6 75 3 91 18 12
Process finished with exit code

```

Задание 1.6

Задача:

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

Список идентификаторов:

Имя	Смысл	Тип
num	Заданное число	int
pointer	Указатель типа char	int
s	Размер заданного числа	char
i	Параметр цикла	int

Код программы:

```

#include <stdio.h>

int main()
{
    int num = 1234567890;
    char *pointer = (char *) &num;
    int s = sizeof(num);
    for (int i = 0; i < s; i++)
    {
        printf("%d ", *(pointer+i));
    }

    return 0;
}

```

Результат выполнения программы:

```
-46 2 -106 73
Process finished with exit code
```

Задание 1.7

Задача:

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

Список идентификаторов:

Имя	Смысл	Тип
matrixPointer	Матрица(массив)	int
m	Количество строк	int
n	Количество столбцов	int
i	Параметр цикла	int
j	Параметр цикла	int

Код программы:

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

int main()
{
    int **matrixPointer = NULL;
    int m;
    int n;
    printf("Enter number of rows");
    scanf("%d", &m);
    printf("Enter number of columns");
    scanf("%d", &n);
    matrixPointer = (int **) malloc(m * sizeof(int *));
    for (int i = 0; i < m; i++) {
        matrixPointer[i] = (int *) malloc(n * sizeof(int));
    }
    for(int i = 0; i < m; i++)
    {
        for(int j = 0; j < n; j++)
        {
            printf("Element[%d][%d]", (i+1), (j+1));
            scanf("%d", &matrixPointer[i][j]);
        }
    }
    for(int i = 0; i < m; i++)
    {
        for(int j = 0; j < n; j++)
        {
            printf("%d ", matrixPointer[i][j]);
        }
    }
    printf("\n");
}
```

```
}  
for (int i = 0; i < m; i++)  
{  
    free(matrixPointer[i]);  
}  
free(matrixPointer);  
  
return 0;  
}
```

Результат выполнения программы:

```
Enter number of rows2  
Enter number of columns2  
Element[1][1]1  
Element[1][2]6  
Element[2][1]47  
Element[2][2]2  
1 6  
47 2  
  
Process finished with exit co
```