

Отчёт №1
Лабораторные работы 1-4

вариант №15

1. лабораторная работа

Разработайте консольное приложение, обеспечивающее ввод с консоли и вывод на консоль строки, целого числа, вещественного числа или символа в соответствии с номером варианта.

В варианте номер 15 было нужно вывести на консоль вещественное число и символ.

```
1  #include<stdio.h>
2
3  int main ()
4  {
5      int a;
6      char sos;
7
8      printf("enter a character: ");
9      scanf("%c", &sos);
10     printf("sos = %c", sos);
11
12     printf("\nenter an integer: ");
13     scanf("%d", &a);
14     printf("a = %d", a);
15
16     return 0;
17 }
```

2. лабораторная работа

Разработайте программу, рассчитывающую значения двух указанных функций. Входные значения аргументов запросите с консоли. Полученные значения функций выведите на консоль.

```
1  #include<stdio.h>
2  #include<math.h>
3
4  int main()
5  {
6      float z1, z2;
7      int b;
8
9      printf("enter an integer\n");
10     scanf("%d", &b);
11
12     z1 = (sqrt(2*b + 2 * sqrt(b * b - 4))) * (1 / (sqrt(b * b - 4) + b + 2));
13     z2 = 1 / (sqrt(b + 2));
14
15     printf("z1 = %f", z1);
16     printf("\nz2 = %f", z2);
17
18     return 0;
19 }
20
```

3. лабораторная работа

1 - Запросите с консоли целое число в указанной системе счисления.

В варианте 15 это - 16-ричная система счисления.

2 - Выведите на консоль введённое число в указанной системе счисления.

В варианте 15 это - 10-чная система счисления.

3 - Выведите на консоль введённое в задании 1 число в 16-ричной или 8-ричной системе счисления, а также это же число в той же системе счисления, но сдвинутое влево/вправо на указанное число бит.

В варианте 15 это - 8-ричная система счисления.

4 - Выведите на консоль введённое в задании 1 число в 16-ричной или 8-ричной системе счисления (согласно заданию 3), а также это же число в той же системе счисления после применения к нему битовой операции отрицания.

5 - Введите с консоли целое число в системе счисления, указанной в задании 3. Выведите на консоль результат указанной битовой операции (и, или, исключающее или) введённого числа и числа, введённого в задании 1 в системе счисления, указанной в задании 3.

```
1  #include<stdio.h>
2  int main() {
3
4      // we are asked to input the number in certain number system
5      int a;
6      printf("enter a hexadecimal value in uppercase - ");
7      scanf("%X", &a);
8      printf("a = %X\n", a);
9
10     // next, we need to transfer hexadecimal into decimal
11     int b;
12     printf("our number in decimal\n");
13     printf("b = %d", a);
14
15     // from hexadecimal to octal, and octal shifted to the right by 3
16     int c;
17     printf("\nnumber in octal\n");
18     printf("c = %o", a);
19
20     printf("\nc shifted is %o\n", a >> 3);
21
22     // output the octal number and octal number with bitwise NOT
23     int d;
24
25     printf("~d = %o\n", ~a);
26
27     // new octal and old octal numbers - operation OR
28     printf("two octal numbers - operation OR %o", a && a);
29     return 0;
30 }
```

4. лабораторная работа

1 - Запросить с консоли целое число и проверить его на попадание в заданный диапазон значений. Результат такой проверки вывести на консоль. При выполнении задания не использовать условный оператор.

В варианте 15 это от -4 до -1.

2 - Запросить с консоли целое число и проверить значение бита с указанным номером в этом числе. Результат такой проверки вывести на консоль. Для удобства разрешается запрашивать указанное число в восьмиричной или шестнадцатиричной системе счисления. При выполнении задания важно обратить внимание на грамотный выбор типа данных анализируемого числа.

В варианте 15 это, соответственно, 15-тый бит.

```
1  #include<stdio.h>
2
3  int main() {
4
5      signed int a;
6      // we create an array with two strings
7      char *m[2] = {"Not in the range", "In the range"};
8      printf("Enter an integer - \n");
9      scanf("%d", &a);
10     // after inputting the integer, we want to make sure that it is in the range, so we check that with ANDing
11     printf("%s\n", m[(a<=-1) && (a>=-4)]);
12     int b;
13     printf("Enter another integer in hexadecimal notation - \n");
14     scanf("%X", &b);
15     // we want to know the 15th bit, so we compare it with a hexadecimal number which has only one 1, in the 15th place
16     int bit = b & 0x4000;
17     //we want to check if the 15th bit is 1 or 0
18     if (bit == 0x4000) {
19         int x = 1;
20         printf("The bit value is %d", x);
21     }
22     else {
23         int x = 0;
24         printf("The bit value is %d", x);
25     }
26
27     return 0;
28 }
29 }
```