Лабораторная №3. Типы данных и операции над ними

Темы, рассматриваемые в лабораторной работе:

A. Работа с целыми числами. Диапазон представления целых чисел. Двоичная арифметика. Использование поразрядных операций.

Создайте проект для следующих реализованных функций.

```
// алгоритм Евклида (геометрический)
int gcd(int a, int b){
//избавляемся от знаков
  a=abs(a);
  b=abs(b);//
  while (a!=b) {
    if (a>b) a-=b;
    else b-=a;
  }
  return a;
}
// примеры использования поразрядных операций
// Вывод на консоль числа в двоичном виде.
// Формат вывода разрядов — от младших к старшим.
void print_bin(short number)
{
      do {
            cout << (number & 1);
            number = number >> 1;
      } while (number);
}
// Вывод на консоль двоичных разрядов числа.
// Формат вывода разрядов — от младших к старшим.
void print_mem_bin(short number) {
      for (int i = 0; i < sizeof(int) * 8; ++i) {
            cout << (number & 1);
            number = number >> 1;
      }
}
// Вывод на двоичных разрядов числа.
// Формат вывода разрядов — от старших к младшим.
void dispBinary (unsigned char u) {
  unsigned char t;
  for (t=128; t>0; t=t>>1)
    if (u&t)
      cout <<"1";
    else
      cout <<"0";
  cout<<"\n";
}

// Эффективная реализация некоторых операций
inline int mul_2(unsigned int x){
  return x<<1;
```

```
}

inline int div_2(unsigned int x){
  return x>>1;
}

inline bool odd (unsigned int x){
  return x&1;
}
```

Проверьте работу функций, используя вызов в функции main.

B. Вещественные типы. Приближенные вычисления.

Создайте проект следующей программы для проверки накопления ошибки в вычислениях с плавающей точкой. Проанализируйте результаты.

```
#include <iostream>
using namespace std;

double sum_1(int n);
double sum_2(int n);

int main() {
  cout.precision(20);
  cout<<sum_1(2000)<<endl;
  cout<<sum_2(2000)<<endl;
  return 0;
}

double sum_1(int n) {
  double s=0;
  n++;    //чтобы учесть слагаемое с номером n
  for (int i=1; i<n; ++i)
    s+= 1.0/i;
  return s;
}
double sum_2(int n) {
  double s=0;
  for (int i=n; i; --i)
    s+= 1.0/i;
  return s;
}
```

Задания:

Для всех реализуемых функций обязательно готовить наборы тестовых заданий.

1. Реализовать функцию проверки (должна возвращать 0 или 1) значения *i*-го бита целого числа.

2. Реализовать функции установки в 1 значения *i*-го бита целого числа, и обнуления значения *i*-го бита целого числа.
3. Реализовать функцию вычисления экспоненты с заданной точностью, с помощью разложения в ряд Тейлора. В тестах использовать сравнение значения, полученного вашей функцией и стандартной функцией exp.
4. Решить аналогичную задачу для синуса угла, заданного радианами. Протестировать функцию, сравнивая значения со стандартной функцией sin. **ВНИМАНИЕ!!!** В тестах к этой задаче обязательно предусмотреть значения отрицательного аргумента и значения аргумента по модулю большего 200. Добиться совпадения результатов и для больших по модулю аргументов.