**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

**Тема №6**

По дисциплине «Основы алгоритмизации и программирования»

На тему «Тестирование кода»

Выполнил:

Студент 1 курса 10 группы ПИ

Макаревич Кирилл Витальевич

Проверил: Белодед Николай Иванович

2024, Минск

**Оглавление**

**1,Введение3**

**2.Описание программы4**

3. Поведение программы5

**3.1 Переполнение при a=a+15**

**3.2 Переполнение при a=a+25**

**3.2 Переполнение при a=a+35**

**4. Объяснение переполнения простым языком 6**

**5. Заключение 7**

**Введение**

В программировании важно учитывать ограничения типов данных, особенно при работе с целочисленными переменными, у которых есть фиксированный диапазон значений. Одной из распространенных ситуаций является переполнение — когда значение переменной выходит за пределы допустимого диапазона. Это приводит к неожиданному поведению программы и может вызвать ошибки. В данном тестировании мы рассмотрим особенности переполнения переменной типа short int в языке программирования C++ и проанализируем поведение программы при выполнении арифметических операций, вызывающих переполнение.

**Описание программы**

Для исследования переполнения используем следующий код на языке C++:

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "ru");  short int a = 32767;  a = a + 1;  cout << "а равно: " << a << endl;  a = a + 2;  cout << "а равно: " << a<<endl;  a = a + 3;  cout << "а равно: " << a;  return 0;  } |

Программа работает с переменной типа short int, которая может хранить целые числа в диапазоне от -32768 до 32767. В начале переменной a присваивается максимальное значение — 32767. После этого к переменной поочередно прибавляются значения, что приводит к переполнению, так как величина переменной выходит за пределы допустимого диапазона.

**Поведение программы**

**1) Переполнение при a = a + 1**

На первом шаге переменной a присваивается значение 32767 — это наибольшее возможное значение для типа short int. При выполнении операции a = a + 1 происходит переполнение, и переменная «сбрасывается» на минимальное значение — -32768. Это связано с тем, что система не может хранить значение больше 32767, и вместо этого начинается отсчет с отрицательного предела диапазона.

Результат вывода программы:



**2) Переполнение при a = a + 2**

После переполнения переменная a равна -32768. При добавлении 2 происходит последовательное увеличение:

* Прибавление 1 к -32768 даст -32767.
* Прибавление еще 1 (второй шаг) даст -32766.

Результат вывода программы:

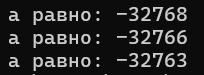


**3) Переполнение при a = a + 3**

На следующем шаге к значению -32766 прибавляется 3:

* Прибавление 1 даст -32765.
* Прибавление еще 1 даст -32764.
* Прибавление еще 1 даст -32763.

Результат вывода программы:



**Объяснение переполнения простым языком**

Чтобы лучше понять механизм переполнения, можно провести аналогию с одометром автомобиля. Представим, что одометр машины(он измеряет пройденный транспортным средством путь) ограничен максимальным значением 99 999 километров. Когда машина достигает этого пробега и проезжает еще 1 километр, одометр не сможет отобразить 100 000 км, так как его предел — 5 разрядов. Вместо этого, он сбрасывается на 0, и отсчет начинается заново.

В контексте переменной short int:

* **Максимальное значение одометра (99 999 км)** аналогично максимальному значению 32767 для переменной типа short int.
* **Сброс на 0** — это аналог переполнения, при котором переменная сбрасывается до -32768, минимального значения диапазона.
* Продолжение движения увеличивает одометр от 0, так же как и переменная после переполнения начинает увеличиваться от минимального значения -32768.

**Заключение**

Переполнение переменных типа short int является важным аспектом, который необходимо учитывать при разработке программ. В данном примере переполнение привело к неожиданному изменению значения переменной при выходе за пределы допустимого диапазона. Аналогия с одометром автомобиля помогает лучше понять этот процесс: как только достигается максимальное значение, система возвращается к минимальному и начинает новый отсчет. Чтобы избежать таких ситуаций, можно использовать типы данных с более широким диапазоном, такие как int или long int, а также применять методы проверки на переполнение перед выполнением арифметических операций.