

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Информатика и вычислительная техника

ОТЧЕТ

**О выполнении лабораторной работы №7.2
«Функции с переменным числом параметров»**

Студент: Шадрин И. Д.

Группа: ИВТ-23-1Б

Преподаватель: Яруллин Д. В.

Пермь – 2024

1. Формулировка индивидуального задания

Написать функцию `sum` с переменным числом параметров, которая находит сумму чисел типа `int` по формуле: $S = a_1 * a_2 + a_3 * a_4 + a_5 * a_6 + \dots$

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных `int`, предназначенный для работы с целыми числами.

3. Исходные коды разработанных программ

Листинг 1: Исходные коды программы main.o (файл: main.cpp)

```
#include <iostream>
#include <cstdint>

using namespace std;

int Sum(int k, ...);

int main()
{
    srand(time(nullptr));

    int arr[12];

    for (int i = 0; i < 12; i++) {
        arr[i] = rand() % 10;
    }

    for (int i = 0; i < 8; i++) {
        cout << arr[i] << " ";

        if (i % 2 == 0) {
            cout << "* ";
        }

        else {
            if (i != 7) {
                cout << "+ ";
            }
        }
    }
    cout << "=" << Sum(8, arr[0], arr[1], arr[2], arr[3], arr[4], arr[5], arr[6],
arr[7]) << endl;

    for (int i = 0; i < 10; i++) {
        cout << arr[i] << " ";

        if (i % 2 == 0) {
            cout << "* ";
        }

        else {
            if (i != 9) {
                cout << "+ ";
            }
        }
    }
    cout << "=" << Sum(10, arr[0], arr[1], arr[2], arr[3], arr[4], arr[5], arr[6],
arr[7], arr[8], arr[9]) << endl;

    for (int i = 0; i < 12; i++) {
```

```

        cout << arr[i] << " ";

        if (i % 2 == 0) {
            cout << "* ";
        }

        else {
            if (i != 11) {
                cout << "+ ";
            }
        }
    }
    cout << "= " << Sum(12, arr[0], arr[1], arr[2], arr[3], arr[4], arr[5], arr[6],
arr[7], arr[8], arr[9], arr[10], arr[11]) << endl;

    return 0;
}

int Sum (int k, ...) {
    va_list argList;
    va_start(argList, k);
    int s = 0;

    for (int i = 0; i < k; i += 2) {
        s += va_arg(argList, int) * va_arg(argList, int);
    }

    va_end(argList);

    return s;
}

```

4. Описание тестовых примеров

Таблица 1: Тестовые примеры

Ввод	Ожидаемый вывод	Вывод
$6 * 4 + 8 * 4 + 1 * 5 + 1 * 4$	65	65
$6 * 4 + 8 * 4 + 1 * 5 + 1 * 4 +$	79	79
$7 * 2$	91	91
$6 * 4 + 8 * 4 + 1 * 5 + 1 * 4 +$		
$7 * 2 + 4 * 3$		
$5 * 1 + 4 * 3 + 5 * 2 + 6 * 4$	51	51
$5 * 1 + 4 * 3 + 5 * 2 + 6 * 4 +$	69	69
$9 * 2$	74	74
$5 * 1 + 4 * 3 + 5 * 2 + 6 * 4 +$		
$9 * 2 + 5 * 1$		
$4 * 1 + 4 * 9 + 3 * 9 + 6 * 7$	109	109
$4 * 1 + 4 * 9 + 3 * 9 + 6 * 7 +$	125	125
$4 * 4$	139	139
$4 * 1 + 4 * 9 + 3 * 9 + 6 * 7 +$		
$4 * 4 + 2 * 7$		

5. Скриншоты

```
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/7.2# g++ -O0 -g3 -o main.o main.cpp
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/7.2# valgrind ./main.o
==31271== Memcheck, a memory error detector
==31271== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==31271== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==31271== Command: ./main.o
==31271==
6 * 4 + 8 * 4 + 1 * 5 + 1 * 4 = 65
6 * 4 + 8 * 4 + 1 * 5 + 1 * 4 + 7 * 2 = 79
6 * 4 + 8 * 4 + 1 * 5 + 1 * 4 + 7 * 2 + 4 * 3 = 91
==31271==
==31271== HEAP SUMMARY:
==31271==   in use at exit: 0 bytes in 0 blocks
==31271==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==31271==
==31271== All heap blocks were freed -- no leaks are possible
==31271==
==31271== For lists of detected and suppressed errors, rerun with: -s
==31271== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/7.2# valgrind ./main.o
==31272== Memcheck, a memory error detector
==31272== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==31272== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==31272== Command: ./main.o
==31272==
5 * 1 + 4 * 3 + 5 * 2 + 6 * 4 = 51
5 * 1 + 4 * 3 + 5 * 2 + 6 * 4 + 9 * 2 = 69
5 * 1 + 4 * 3 + 5 * 2 + 6 * 4 + 9 * 2 + 5 * 1 = 74
==31272==
==31272== HEAP SUMMARY:
==31272==   in use at exit: 0 bytes in 0 blocks
==31272==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==31272==
==31272== All heap blocks were freed -- no leaks are possible
==31272==
==31272== For lists of detected and suppressed errors, rerun with: -s
==31272== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/7.2# valgrind ./main.o
==31273== Memcheck, a memory error detector
==31273== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==31273== Using Valgrind-3.16.1 and LibVEX; rerun with -h for copyright info
==31273== Command: ./main.o
==31273==
4 * 1 + 4 * 9 + 3 * 9 + 6 * 7 = 109
4 * 1 + 4 * 9 + 3 * 9 + 6 * 7 + 4 * 4 = 125
4 * 1 + 4 * 9 + 3 * 9 + 6 * 7 + 4 * 4 + 2 * 7 = 139
==31273==
==31273== HEAP SUMMARY:
==31273==   in use at exit: 0 bytes in 0 blocks
==31273==   total heap usage: 2 allocs, 2 frees, 73,728 bytes allocated
==31273==
==31273== All heap blocks were freed -- no leaks are possible
==31273==
==31273== For lists of detected and suppressed errors, rerun with: -s
==31273== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
root@aw313146638:~/labs/Labs_PSTU/Sem_2/Green/7.2#
```

Рис. 1: Сборка и запуск программы main.o с помощью valgrind