Лабораторная работа номер 2
Вариант 5

Сделал: Каравка Денис Иванович
Группа: 1БПМИ-УДМО

Постановка задачи:

Var x:array [1..9999] of real; s:real;

Вычислить (индекс 1-ого слагаемого каждой суммы-квадрат):

s=(x1+x2+x3) (x4+x5+…+x8) (x9+ … +x15)…(x9801+…+x9999).

Исходные данные:

x1,…,x9999 – массив действительных чисел

size – размерность массива (Равна 9999)

Результаты выполнения алгоритма:

result – произведение суммы-квадратов

Математическая формула условия задачи

1) result = (x1+x2+x3)*(x4+x5+…+x8)*(x9+ … +x15)…(x9801+…+x9999).

Описание свойств модулей, используемых в программе

В данной программе используется функция подсчета сумм-квадратов

Свойства модуля:

Текст функции:

```
int summarySquare(int begin, int square, int number[]){
  int summary = 0;
  for(int i = begin; i <= square - 1; i++){
    summary += number[i];
  }
  cout << "Сумма " << begin << " - " << square-1 << " элементов: " << summary <<
endl;
  return summary;
}
```

Размер функции: 6 строк

Прочность функции: функционально прочный модуль, поскольку в данной функции реализуется одна функция, а именно подсчет суммы

Сцепление с другими модулями: параметрическое сцепление, так как данные передаются при обращении к нему в виде входных параметров begin, square, number[] и результат (выходной параметр summary) его выполнения в программе передается переменной.

Рутинность модуля: данный модуль является рутинным, так как его результат не зависит от предыстории обращений к нему, то есть каждый результат его выполнения является независимым от предыдущих

Программный код

Программа была написана на языке программирования C++

```
#include <iostream>
#include <math.h>
using namespace std;
/*
Var x:array [1..9999] of real; s:real;
Вычислить (индекс 1-ого слагаемого каждой суммы-квадрат):
s=(x1+x2+x3) (x4+x5+…+x8) (x9+ … +x15)…(x9801+…+x9999).
*/
int summarySquare(int begin, int square, int number[]){
  int summary = 0;
  for(int i = begin; i <= square - 1; i++){
    summary += number[i];
  }
  cout << "Сумма " << begin << " - " << square-1 << " элементов: " << summary <<
endl;
  return summary;
}
int main()
{
  int size = 9999;
  int n[size];
  int square = 1;
  long long result = 1;
  for(int i = 0; i <= size; i++){
    n[i] = 1;
  }
  int i = 1;
  int j = 2;
  while(i < size){
    square = j*j;
    result *= summarySquare(i, square, n);
    i = square;
    j ++;
  }
  cout << "Произведение всех элементов: " << result;
}
```

Тестирование программы

Тестовые данные номер 1:

size = 16; x1…x16 = 1;

Ожидаемый результат:

result = (1+1+1)(1+1+1+1+1)(1+1+1+1+1+1+1) = 3*5*7 = 105;

Тестовые данные номер 2:

size = 25; x1…x25 = 1;

Ожидаемый результат:

result = (1+1+1)(1+1+1+1+1)(1+1+1+1+1+1+1)(1+1+1+1+1+1+1+1+1) = 3*5*7*9 = 945;