**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 18

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели на функции»

Выполнила:  
Студентка 1 курса, 10 группа  
Кириленко Ангелина Васильевна  
Преподаватель: асс. Харланович А.В

2024 г. Минск

|  |  |
| --- | --- |
| **Задание** | **Выполнение задания** |
| 2. В программе, приведенной в правой части, демонстрируется использование *указателя на функцию* для вычисления площади криволинейной трапеции | #include <iostream> using namespace std;  float integ(float(\*fun)(float), float a, float b, float h);  float f(float x);  int main()  {  float z;  z = integ(f, 0.0f, 10.0f, 0.01f);  cout << "Result=" << z << endl;  return 0;  }  float integ(float(\*fun)(float), float a, float b, float h)  {  float x, s = 0.0f;  x = a + h;  while (x <= b)  {  s += h \* fun(x);  x = x + h;  }  return s;  }  float f(float x)  {  return (2 \* x + 5);  } |
| 3. Выполнить программу, содержащую функцию с *переменным числом параметров*. Записать условие задачи. | Условие:  #include <iostream> using namespace std;  int sum(int n, ...);  using namespace std;  void main()  {  cout << sum(6, 4, 5, 1, 2, 3, 0) << endl;  cout << sum(2, 34, 4445);  }  int sum(int n, ...)  {  int\* p = &n;  int sum = 0;  for (int i = 1; i <= n; i++)  sum += \*(++p);  return sum;  }  Напишите программу, которая вычисляет сумму произвольного количества целых чисел, переданных в качестве аргументов. |
| 4. В соответствии со своим вариантом *отделить корни* двух уравнений и вычислить их методом *дихотомии* для исходных данных из таблицы, приведенной ниже. Точность вычислений принять равной **e** = 0,001 для всех вариантов. Операторы метода вычисления корня оформить в виде функции пользователя, уравнения записать также в виде функций пользователя. В основной программе предусмотреть ввод исходных данных, обращение к функции, реализующей метод дихотомии. В процессе выполнения программы определить корни двух уравнений. Использовать *указатель на функцию.*  Вариант 8  5. В соответствии со своим вариантом написать программы по условиям задач из таблицы ниже. Программа должна содержать функцию пользователя с *переменным числом параметров* и не менее трех обращений к ней с различным количеством параметров.  **Дополнительные задания**  Вариант 16 | #include <iostream>  #include <cmath>  using namespace std;  float metod\_dihotomii(float (\*fun)(float), float a, float b, float e) {  float x;  do {  x = (a + b) / 2;  if (fun(x) \* fun(a) <= 0) {  b = x;  }  else {  a = x;  }  } while (abs(a - b) > 2 \* e);  return x;  }  float f1(float x) {  return pow(x, 3) + 3 \* x - 1;  }  float f2(float x) {  return exp(x) - 4;  }  void function() {  float e = 0.001, a, b;  cout << "Введите a: ";  cin >> a;  cout << "Введите b: ";  cin >> b;  cout << endl;  cout << "Корень первого уравнения: " << metod\_dihotomii(f1, a, b, e) << endl;  cout << "Корень второго уравнения: " << metod\_dihotomii(f2, a, b, e) << endl;  }  int main() {  setlocale(LC\_ALL, "Russian");  function();  return 0;  }    #include <iostream>  #include <cstdarg>  using namespace std;  void mn(int n, ...) {  va\_list arg;  va\_start(arg, n);  int min = va\_arg(arg, int);  for (int i = 1; i < n; i++) {  int val;  val = va\_arg(arg, int);  if (val < min) {  min = val;  }  }  cout << "Минимальное значение: " << min << endl;  va\_end(arg);  }  int main() {  setlocale(LC\_ALL, "rus");  mn(5, 23, 1, 4, 3, 9);  mn(4, 15, 3, 2, 12);  mn(6, 320, 52, 148, 22, 34, 30); }    #include <iostream> #include <string>  using namespace std;  int countWordsStartingWithA(const string& text) {  int count = 0;  bool inWord = false;  for (size\_t i = 0; i < text.size(); ++i) {  if (text[i] != ' ' && !inWord) {  inWord = true; // начало нового слова  if (text[i] == 'а' || text[i] == 'А') { // проверяем первую букву слова  ++count;  }  }  else if (text[i] == ' ') {  inWord = false; // конец слова  }  }  return count;  }  int main() {  setlocale(LC\_ALL, "Russian");  string lines[] = { "Арбуз ананас абрикос банан", "Апельсин лимон киви яблоко", "Тут текст только", "АБВ агд ежз аик" };  for (const string& line : lines) {  int count = countWordsStartingWithA(line);  cout << "В строчке: \"" << line << "\", здесь " << count << " слова начинаются с 'а'.\n";  }  return 0;  } |
| Вариант 12  Вариант 4 | #include <iostream> #include <vector>  #include <algorithm>  using namespace std;  void processArray(int\* arr, int n) {  int pos = 0;  for (int i = 0; i < n; ++i) {  if (arr[i] < 0) {  arr[i] = 0;  }  else {  if (i > pos) {  swap(arr[i], arr[pos]);  }  pos++;  }  }  }  void printArray(int\* arr, int n) {  for (int i = 0; i < n; ++i) {  cout << arr[i] << " ";  }  cout << endl;  }  int main() {  setlocale(LC\_ALL, "Russian");  int arr1[] = { 23, -1, 4, -3, 9, -5 };  int n1 = sizeof(arr1) / sizeof(arr1[0]);  cout << "Исходный массив 1: ";  printArray(arr1, n1);  processArray(arr1, n1);  cout << "Массив 1 после обработки: ";  printArray(arr1, n1);  int arr2[] = { 15, 3, -2, 12, -8, -1, 0, 5, -2 };  int n2 = sizeof(arr2) / sizeof(arr2[0]);  cout << "Исходный массив 2: ";  printArray(arr2, n2);  processArray(arr2, n2);  cout << "Массив 2 после обработки: ";  printArray(arr2, n2);  int arr3[] = { 10, 20, 30, 40, 50 };  int n3 = sizeof(arr3) / sizeof(arr3[0]);  cout << "Исходный массив 3: ";  printArray(arr3, n3);  processArray(arr3, n3);  cout << "Массив 3 после обработки: ";  printArray(arr3, n3);  int arr4[] = { -1, -2, -3 };  int n4 = sizeof(arr4) / sizeof(arr4[0]);  cout << "Исходный массив 4: ";  printArray(arr4, n4);  processArray(arr4, n4);  cout << "Массив 4 после обработки: ";  printArray(arr4, n4);  return 0; }    #include <iostream> #include <cstdarg>  using namespace std;  void mn(int n, ...) {  va\_list arg;  va\_start(arg, n);  int max = va\_arg(arg, int);  for (int i = 1; i < n; i++) {  int val;  val = va\_arg(arg, int);  if (val > max) {  max = val;  }  }  cout << "Максимальное значение: " << max << endl;  va\_end(arg);  }  int main() {  setlocale(LC\_ALL, "Russian");  mn(5, 10, 12, 4, 3, 9);  mn(4, 15, 3, 2, 12);  mn(6, 320, 52, 148, 22, 34, 30);  } |