МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ О ВЫПОЛНЕНИИ

ПРАКТИЧЕСКОЙ РАБОТЫ

«Оператор switch. Текстовое меню. Циклы. Табулирование кусочной функции»

студента 2 курса, ИСП-207 группы Асылбек уулу Бакыта

Направление 09.02.07 – «Информационные системы и программирование»

Руководитель:

преподаватель

М.В. Валеев

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

Москва 2022

Оглавление

[1. Постановка задачи 3](#_Toc183863382)

[2. Структурные требования 3](#_Toc183863383)

[3. Описание задачи 3](#_Toc183863384)

[3.1. Используемые переменные 3](#_Toc183863385)

[3.2. Псевдокод 4](#_Toc183863386)

[3.3. Блок-схема 6](#_Toc183863387)

[3.4. Листинг программной реализации 10](#_Toc183863388)

[3.5. Результат 12](#_Toc183863389)

[СПИСОК ЛИТЕРАТУРЫ 14](#_Toc183863390)

# Постановка задачи

1. Для задачи табулирования кусочной функции разработайте программу с пользовательским интерфейсом. Программа должна иметь пользовательский интерфейс – текстовое меню, в пунктах которого предусмотреть следующие действия:

* ввод исходных данных (начало, конец интервала, шаг приращения аргумента);
* табулирование функции циклом while;
* табулирование функции циклом do...while;
* табулирование функции циклом for;
* выход и завершение работы программы.

1. Начало, конец отрезка табулирования и шаг – произвольные, вводятся с клавиатуры. Вычисление функции должно происходить внутри пользовательской функции с возвратом результата (оператор return).
2. Составьте таблицу значений функции y = F(x) согласно индивидуальному варианту (см. ниже). Все результаты вычисления отобразить на экране. Вычисление функции должно происходить внутри пользовательской функции с возвратом результата для вывода на экран. Вычисление выполнить с использованием условного оператора if (...) ...; или if (...) ... else ...;

Вариант 1

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Условие | Константы | Диапазон и шаг изменения |
|  |  |  |  |

# Структурные требования

Отсутствуют

# 3. Описание задачи

## 3.1. Используемые переменные

y – переменная, управляющая работой оператора switch;

a – константа;

b – константа;

c – переменная, которую надо вычислить;

t – аргумент;

## 3.2. Псевдокод

Начало

Вывод: \* Организация программного меню \*

Объявление a, b, t, y

Пока

Вывод: МЕНЮ

Вывод: 1 - Задача 1

Вывод: 2 - Задача 2

Вывод: 3 – Задача 3

Вывод: 4 – Выход

Вывод: Какую задачу выберите?

Ввод: y

Переключатель(y)

Выбор 1

Пока (t < 3)

если (t < 1)

Вывод: с

Иначе

если

Вывод: с

Иначе

Если (t > 2)

Вывод: с

Остановить

Выбор 2

Делать

если (t < 1)

Вывод: с

Иначе

если

Вывод: с

Иначе

Если (t > 2)

Вывод: с

Пока (t < 3)

Остановить

Выбор 3

Для ( )

если (t < 1)

Вывод: с

Иначе

если

Вывод: с

Иначе

Если (t > 2)

Вывод: с

Остановить

Выбор 4

Вывод: Работа программы завершается...

Остановить

Конец

## 3.3. Блок-схема

Начало

Вывод: \*Организация программного меню \*

Вывод: меню 1 - Задача 1 2 - Задача 2 3 - Задача 3 4 – Выход Какую задачу выберите

Ввод: y

1

1 <= t <= 2

Вывод: с

2

1 <= t <= 2

Вывод: с

3

1 <= t <= 2

Вывод: с

4

Вывод: Работа программы завершается...

Конец

## 3.4. Листинг программной реализации

#include <stdio.h>  
#include <math.h>  
#include <windows.h>  
  
int main() {  
 SetConsoleOutputCP(**CP\_UTF8**);  
 printf("\* Организация программного меню \*\n");  
 int y = 0; *// переменная, управляющая работой оператора switch* float a = -0.5, b = 2, t, c;  
 while (y != 4) {  
 printf("\n МЕНЮ \n1 - Задача 1\n2 - Задача 2\n3 - Задача 3\n4 - Выход\nКакую задачу выберите\n");  
 scanf("%i", &y);  
 switch (y) {  
 case 1:  
 t = 0;  
 while (t < 3) {  
 t += 0.15;  
 if (t < 1) {  
 c = 1;  
 printf("%f\n", c);  
 } else {  
 if (1 <= t && t <= 2) {  
 c = a \* pow(t, 2) \* log(t);  
 printf("%f\n", c);  
 } else {  
 if (t > 2) {  
 c = exp(a \* t) \* cos(b \* t);  
 printf("%f\n", c);  
 }  
 }  
 }  
 }  
 break;  
 case 2:  
 t = 0;  
 do {  
 t += 0.15;  
 if (t < 1) {  
 c = 1;  
 printf("%f\n", c);  
 } else {  
 if (1 <= t && t <= 2) {  
 c = a \* pow(t, 2) \* log(t);  
 printf("%f\n", c);  
 } else {  
 if (t > 2) {  
 c = exp(a \* t) \* cos(b \* t);  
 printf("%f\n", c);  
 }  
 }  
 }  
 } while (t < 3);  
 break;  
 case 3:  
 for (t = 0; t < 3; t += 0.15) {  
 if (t < 1) {  
 c = 1;  
 printf("%f\n", c);  
 } else {  
 if (1 <= t && t <= 2) {  
 c = a \* pow(t, 2) \* log(t);  
 printf("%f\n", c);  
 } else {  
 if (t > 2) {  
 c = exp(a \* t) \* cos(b \* t);  
 printf("%f\n", c);  
 }  
 }  
 }  
 }  
 break;  
 case 4:  
 printf("Работа программы завершается...\n");  
 break;  
 default:  
 printf("\n Вы ввели неверное значение.\nПовторите ввод\n");  
 break;  
 } *// конец оператора switch* } *// конец оператора цикла while* return 0;  
}

## 3.5. Результат

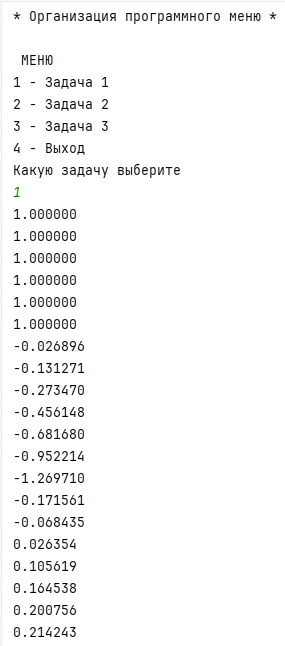


Рисунок – 1

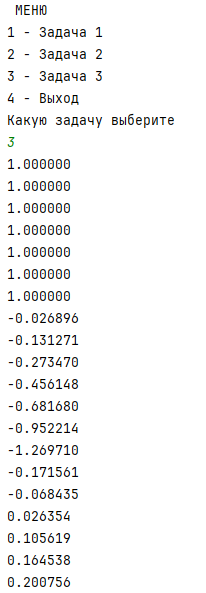
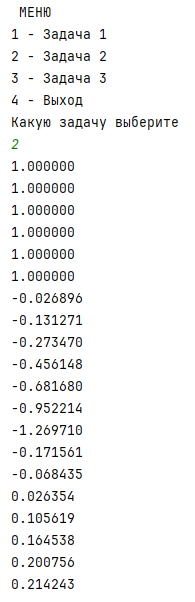


Рисунок – 2 Рисунок – 3

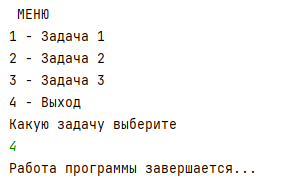


Рисунок – 4

# СПИСОК ЛИТЕРАТУРЫ

1. Конова Е.А., Поллак Г.А. Алгоритмы и программы. Язык С++: Учебное

пособие. - 2-е изд., стер. - СПб.: Издательство "Лань", 2017. - 384 с.

2. Седжвик Роберт. Алгоритмы на С++.: Пер. с англ. - М.: ООО "И.Д. Вильямс",

2011. - 1056 с.

3. Лафоре Р. Объектно-ориентированное программирование в С++. Классика

Computer Science. 4-е изд. - СПб.: Питер, 2015. - 928 с.

4. Орлов С.А. Теория и практика языков программирования: Учебник для вузов.

Стандарт 3-го поколения. - СПб.: Питер, 2014. - 688 с.

5. Павловская Т.А. С/С++. Процедурное и объектно-ориентированное

программирование: Учебник для вузов. Стандарт 3-го поколения. - СПб.:

Питер, 2015. - 496 с.

6. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник

для вузов. - СПб.: Питер, 2014. - 432 с.

7. Плаксин М.А. Тестирование и отладка программ - для профессионалов

будущих и настоящих. - М.: БИНОМ. Лаборатория знаний, 2013. - 167 с.