МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по практическим работам  
по дисциплине «Основы алгоритмизации и программирования»

Выполнил Проверил

студент группы ГИ-11 ст. преп. каф. ИУСиТ

Зайцев И. С. Голдобина Т. А.

Гомель, 2019

Содержание

[Практическая работа №6 Разработка алгоритма, составление, отладка и выполнение программы с ветвлением 3](#_Toc25224552)

[Задания 3](#_Toc25224553)

[Контрольные вопросы 9](#_Toc25224554)

# Практическая работа №6 Разработка алгоритма, составление, отладка и выполнение программы с ветвлением

### **Цель**

Разработать алгоритм и составить, отладить и выполнить программы с ветвлением.

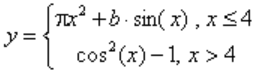
## Задания

### **Задание 1**

### Условие

### Изобразить блок-схему алгоритма и составить программу, вычисляющую значение выражения:

### Программный код

  
Рисунок 1 – Условие задания 1

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define PI 3.1415926

int main()

{int x, b;

double y;

scanf("%d%d",&x,&b);

if (x<=4){

y=PI\*pow(x,2)+ b \* sin(x);}

else {

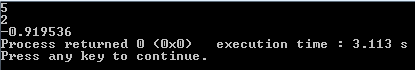
y=pow(cos(x),2)-1;}

printf("%f",y);

return 0;

}

### **Блок-схема:**

  
Рисунок 2 – Решение задания 1

### 

### Задание 2

### Условие

Составить блок-схему алгоритма и программу, в которой вычисляется значение выражения:

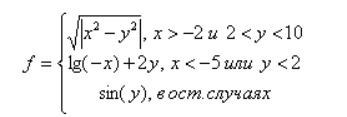


Рисунок 3 - Условие задания 2

**Программный код**

#include <stdio.h>

#include <stdlib.h>

int main()

{int x, y;

double f;

scanf("%d%d",&x, &y);

if (x>-2 && 2<y && y<10){

f= sqrt(abs(pow(x,2)-pow(y,2)));}

else if (x<-5 || y<2){

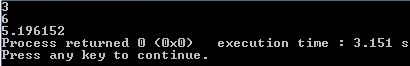
f=log10(-x)+ 2 \* y;}

else {

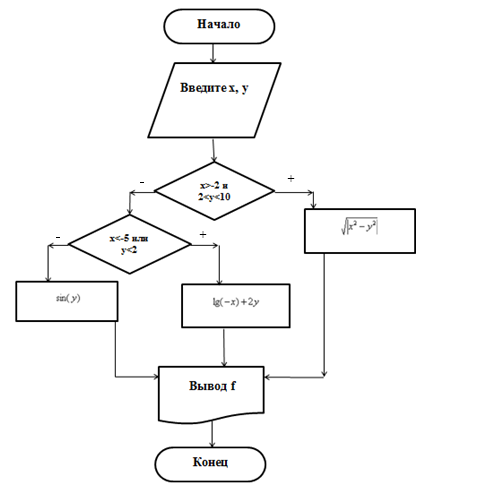
f=sin(y);}

printf("f = %f", f);

return 0;}

  
Рисунок 3 – Решение задания 2

### **Блок-схема:**



### Задание 3

**Условие**

Составить блок-схему алгоритма и программу для решения задачи:

Ввести номер четверти декартовой системы координат. Напечатать неравенства, задающие область значений x и y. Например, для 1-й четверти, "x > 0, y > 0". Предусмотреть возможность некорректного ввода исходных данных

**Программный код**

#include <stdio.h>

#include <stdlib.h>

int main()

{int x;

scanf("%d",&x);

switch (x){

case 1:

printf("x>0, y>0");

break;

case 2:

printf("x<0, y>0");

break;

case 3:

printf("x<0, y<0");

break;

case 4:

printf("x>0, y<0");

break;

default:

printf("NOT FOUND");

break;}

return 0;

}

  
Рисунок 4 – Решение задания 3

Блок-схема:

### 

### Задание 4

**Условие**

### Составить программу, в которой для решения задачи из задания 1 используется тернарный оператор.

**Программный код**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define PI 3.1415926

int main()

{int x, b;

double y;

scanf("%d%d",&x,&b);

y=(x<=4) ? PI\*pow(x,2)+ b \* sin(x) : pow(cos(x),2)-1;

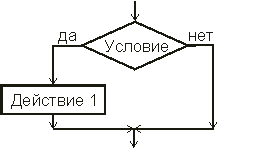
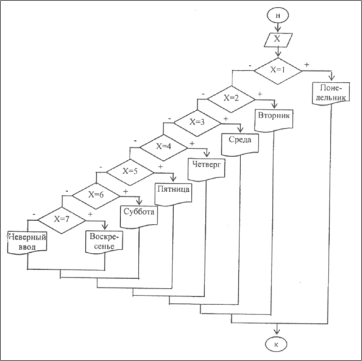
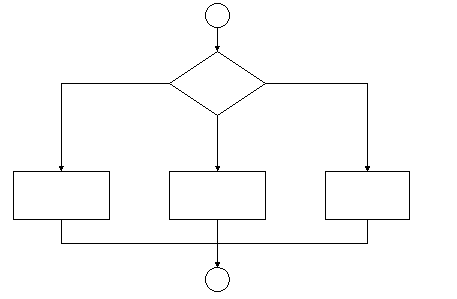
printf("%f",y);

return 0;}



Рисунок 5 - Решение задания 4

## Контрольные вопросы

1. Разветвляющиеся алгоритмы (ветвление, альтернатива) - Это алгоритмы, в которых естественное линейное следование не является обязательным
2. Блок-схема разветвляющейся вычислительной конструкции – В задании №3
3. Неполное ветвление. Понятие и блок-схема - Ветвление при выполнении определённого условия пропускается некоторый блок операций. 
4. Вложенное ветвление. Понятие и блок-схема - Любая ветвь может не быть линейным участком программы, а сама содержать ветвление. 
5. Множественное ветвление. Понятие и блок-схема - ветвление, с более чем двумя вариантами действий. 
6. Оператор ветвления в *C*. Общая форма и принцип действия - if(a==b) {

}

else {

}

Если условие в скобках после if верно, выполняется код в первых фигурных скобках. А если не верно, то во вторых скобках.

1. Неполное ветвление в C. Формат и принцип действия.

if(a==b) {

}

Если условие в скобках после if верно, выполняется код в фигурных скобках.

1. Вложенное ветвление в C. Формат и принцип действия

if(x==y) {

if(c==y) {}

else {}}

else {}

1. Множественное ветвление в C. Формат и принцип действия - Логика выполняющейся программы может быть сложнее, чем выбор одной из двух ветвей. Например, в зависимости от значения той или иной переменной, может выполняться одна из трех (или более) ветвей программы –

Switch(z){

Case 1:

//

Break;

Case 2:

//

Break;

Default:

//

Break;}

### **Выводы по работе**

Разработал алгоритм, составил, отладил и выполнил программы с ветвлением.