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

Оглавление

[Практическая работа №4 Разработка алгоритма, составление, отладка и выполнение программы линейного вычислительного процесса 3](#_Toc22569573)

[1.1 Задания 3](#_Toc22569574)

[1.2 Контрольные вопросы 7](#_Toc22569575)

# Практическая работа №4 Разработка алгоритма, составление, отладка и выполнение программы линейного вычислительного процесса

### **Цель**

Разработать алгоритм и выполнить программы линейного вычислительного процесса.

## 1.1 Задания

### **Задание 1**

### Условие

Составить блок-схему алгоритма и линейную программу, которая позволяет ввести значения переменных и вычислить указанные выражения по формулам.

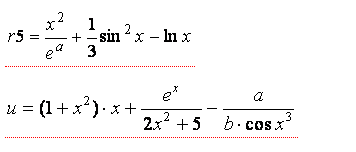


Рисунок 1 – Фрагмент программного кода

### 

Рисунок – Блок-схема

### Программный код

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define e 2,71828

int main()

{int x, a, b;

float r5, u;

printf("Enter\n");

printf("x=");

scanf("%d",&x);

printf("a=");

scanf("%d",&a);

printf("b=");

scanf("%d",&b);

r5=(pow(x,2)/exp(a))+ 1/3.\*pow(sin(x),2)-log(x);

u=((1+(pow(x,2)))\*x)+exp(x)/(2\*pow(x,2)+5)- a/b\*cos(pow(x,3));

printf("r5=");

printf("%f\n",r5);

printf("u=");

printf("%f",u);

return 0;

}

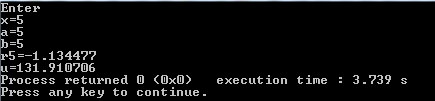


Рисунок 3 – Результат задания 1

### **Задание** 2

### Условие

Написать программу, которая запрашивает 8 целых чисел в следующем формате:

43 7 17 34 8 34 3 12

Для считывания используется единственная функция scanf(). При этом считываются 1-е, 3-е, 5-е и 6-е значения и соответственно записываются в переменные**oh, co, m, u.**  
Определить с помощью возвращаемого значения функции scanf() и вывести на экран количество успешно считанных значений.

Вывести на экран с помощью функции printf() значение переменной**oh**три раза: до применения, с применением операции ***префиксной формы инкремента*** и после нее.  
Вывести на экран с помощью функции printf() значение переменной**co**три раза: до применения, с применением операции ***постфиксной формы инкремента*** и после нее.  
Вывести на экран с помощью функции printf() значение переменной**m**три раза: до применения, с применением операции ***префиксной формы декремента*** и после нее.  
Вывести на экран с помощью функции printf() значение переменной**u**три раза: до применения, с применением операции ***постфиксной формы декремента*** и после нее.

### Программный код

#include <stdio.h>

#include <stdlib.h>

int main()

{int oh, co, m, u, x;

x=scanf("%d %\*d %d %\*d %d %d %\*d %\*d",&oh, &co, &m, &u);

printf ("%d %d %d %d\n",oh,co,m,u);

printf("x=%d\n",x);

printf ("oh=%d\n",oh);

printf ("oh=%d\n",++oh);

printf ("oh=%d\n\n",oh);

printf ("co=%d\n",co);

printf ("co=%d\n",co++);

printf ("co=%d\n\n",co);

printf ("m=%d\n",m);

printf ("m=%d\n",--m);

printf ("m=%d\n\n",m);

printf ("u=%d\n",u);

printf ("u=%d\n",u--);

printf ("u=%d\n\n",u);

return 0;

}

Рисунок 4 - Результат задания 2

### Задание 3

### Условие

Написать программу, которая запрашивает 7 целых чисел в следующем формате:

34!7@ 27;3 98:34 #9

Для считывания используется единственная функция scanf(). При этом считываются 1-е, 2-е, 4-е и 7-е значения и соответственно записываются в переменные**co, me, h, ere.**  
Определить с помощью возвращаемого значения функции scanf() и вывести на экран количество успешно считанных значений.  
Вывести значения всех считанных переменных на экран.  
С помощью сокращенной операции вычитания из переменной **me**вычесть значение переменной**h.**  
Поменять местами значения переменных**co**и**ere.**  
Вывести новые значения переменных на экран.

### Программный код

#include <stdio.h>

#include <stdlib.h>

int main()

{int co, me, h, ere, x, p;

x=scanf("%2d%\*[!]%1d%\*[@ 27 ;]%1d %\*[98]%\*[:, , 34, #]%1d",&co, &me, &h, &ere);

printf("%d %d %d %d\n", co,me,h,ere);

printf("x=%d\n",x);

printf("me-h=%d\n", me-h);

p=co;

co=ere;

ere=p;

printf("%d %d", co, ere);

return 0;}

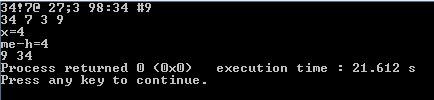


Рисунок 5 - Результат задания 3

## Контрольные вопросы

1. Способы описания алгоритмов.  
Схематические. Псевдокод. Текстуальные.  
2. Графический способ описания алгоритмов.  
Схема программы отображает последовательность операций в программе и включает символы данных, символы процесса, линейные символы и специальные символы.  
3. Основные символы данных.

Параллелограмм - обозначает вводимые данные. Прямоугольник с волнистой нижней стороной - используется для обозначения выводимых данных.  
4. Блок, используемый для вывода информации в документ.  
Символ данных «Документ», представленный в виде прямоугольника с волнистой нижней стороной.  
5. Символ процесса.  
Прямоугольник, который используется для обозначения обработки данных любого типа.  
6. Как перенести часть блок-схемы на новую страницу?  
С помощью специального символа «Соединитель», представленного в виде окружности.  
7. Спецификатор %p.  
Используется для ввода или вывода указателя.  
8. Спецификатор %n вывода данных.  
Используется для определения числа символов, выведенных к моменту обработки самого спецификатора.   
9. Спецификатор %n ввода данных.  
Используется для определения количества прочитанных символов.  
10. Использование набора сканируемых символов.  
%[] используется для определения набора символов, которые могут быть прочитаны функцией scanf() и присвоены соответствующему массиву символов.

### **Выводы по работе**

Составил блок-схему. Написал линейные программы которые позволяют ввести значения переменных и вычислить указанные выражения по формулам.