Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №9**

По дисциплине «Основы алгоритмизации и программирования»

Тема: «Пользовательские функции»

**Выполнил:**

Студент 1 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Гирель Т.Н

Брест 2021

**Цель работы:** Изучить основные принципы написания пользовательских функций, ознакомиться с возможностями передачи данных в функции и получения результата по итогам работы функции. Реализовать собственные функции для обработки данных составных и простых типов.

**Вариант №1**

**Текст задания №1**

**Описать функцию MinInt(A, N) целого типа, находящую минимальный элемент целочисленного массива A размера N. С помощью этой функции найти минимальные элементы массивов A, B, C размера NA, NB, NC соответственно.**

**Текст программы**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

int MinInt(int\* a,int n){

int min=\*a,i;

for(i=0;i<n;i++){

if(min>\*a) min=\*a;

a++;}

return min;}

int main(){

srand(time(NULL));

int n;

printf("Vedite Razmer: ");

scanf("%d",&n);

int a[n],b[n],c[n];

for(int i=0;i<n;i++){

a[i]= rand()%(99);

printf("a[%d] = %d ",i,a[i]);}

printf("\nMinInt:%d\n",MinInt(a,n));

for(int i=0;i<n;i++){

b[i]= rand()%(10-0+1);

printf("b[%d] = %d ",i,b[i]);}

printf("\nMinInt:%d\n",MinInt(b,n));

for(int i=0;i<n;i++){

c[i]= rand()%(10-0+1);

printf("c[%d] = %d ",i,c[i]);}

printf("\nMinInt:%d\n",MinInt(c,n));

return 0;

}

**Результат работы**

Vedite Razmer: 5

a[0] = 41 a[1] = 65 a[2] = 64 a[3] = 68 a[4] = 37

MinInt:37

b[0] = 9 b[1] = 7 b[2] = 7 b[3] = 1 b[4] = 2

MinInt:1

c[0] = 9 c[1] = 7 c[2] = 10 c[3] = 4 c[4] = 4

MinInt:4

**Проверка корректности программы**

|  |  |  |
| --- | --- | --- |
| Входные данные | Промежуточное данные | Выходные данные |
| 5 | А = [41,65,64,68,37]  B = [9,7,7,1,2]  C = [9,7,10,4,4] | 37  1  4 |

**Текст задания №2**

**Описать функцию IsIdent(S) целого типа, проверяющую, является ли строка S допустимым идентификатором, то есть непустой строкой, которая содержит только латинские буквы, цифры и символ подчеркивания «\_» и не начинается с цифры. Если S является допустимым идентификатором, то функция возвращает 0. Если S является пустой строкой, то возвращается –1, если S начинается с цифры, то возвращается –2. Если S содержит недопустимые символы, то возвращается номер первого недопустимого символа. Проверить с помощью функции IsIdent пять данных строк.**

**Текст программы**

#include <stdio.h>

#include <stdlib.h>

int IsDent(char\* s){

if(\*s=='\0') return -1;

if((int)\*s>=48 && (int)\*s<=57) return -2;

int nomer=1;

while (\*s!='\0'){

if(((int)\*s>=48&&(int)\*s<=57)||

((int)\*s>=65&&(int)\*s<=90)||

((int)\*s>=97&&(int)\*s<=122)||

(int)\*s==95){s++;nomer++;}

else return nomer;}return 0;}

int main(){

char s1[99],s2[99],s3[99],s4[99],s5[99];

gets(s1);gets(s2);gets(s3);gets(s4);gets(s5);

printf("\nIsDent:%d",IsDent(s1));

printf("\nIsDent:%d",IsDent(s2));

printf("\nIsDent:%d",IsDent(s3));

printf("\nIsDent:%d",IsDent(s4));

printf("\nIsDent:%d",IsDent(s5));return 0;}

**Результат работы**

1asdv

\_name

sama%f

no\_mer

IsDent:-2

IsDent:0

IsDent:-1

IsDent:5

IsDent:0

**Проверка корректности программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1asdv  \_name  sama%f  no\_mer | -2  0  -1  5  0 |

**Текст задания №3**

**Описать рекурсивную функцию Fact(N) целого типа, вычисляющую значение факториала**

**N! = 1·2·...·N**

**(N > 0 — параметр целого типа). С помощью этой функции вычислить факториалы пяти данных чисел.**

**Текст программы**

#include <stdio.h>

int Fact(int n){

if(n==0||n==1) return 1;

return n\*Fact(n-1);}

int main(){

int n;

for(int i=0;i<5;i++){

printf("Vedite chislo: ");

scanf("%d",&n);

printf("Factorial !%d = %d\n",n,Fact(n));}

return 0;}

**Результат работы**

Vedite chislo: 5

Factorial !5 = 120

Vedite chislo: 6

Factorial !6 = 720

Vedite chislo: 7

Factorial !7 = 5040

Vedite chislo: 8

Factorial !8 = 40320

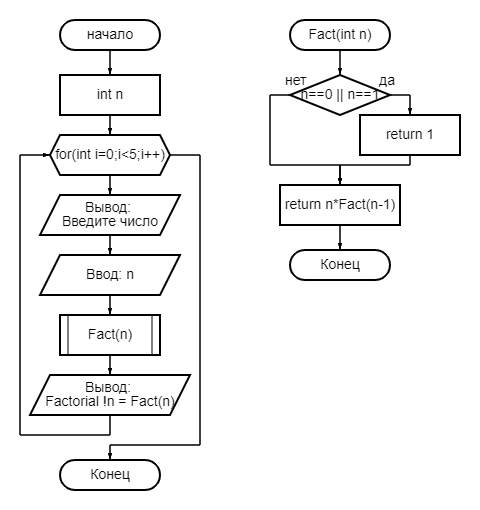
Vedite chislo: 9

Factorial !9 = 362880

**Проверка корректности программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5  6  7  8  9 | 120  720  5040  40320  362880 |

**Блок-схема алгоритма**

****

**Вывод:** Изучил основные принципы написания пользовательских функций, ознакомился с возможностями передачи данных в функции и получения результата по итогам работы функции. Реализовал собственные функции для обработки данных составных и простых типов.