САА - Упражнение 3 (Лаб\_3\_САА)

Задача 21\*

Съставете алгоритъм и напишете програма, която намира броя на двойките елементи симетрични на главния диагонал на двумерен масив, в които елементът над главния диагонал е по-малък от елементът под главния диагонал. Например, следните двойки от елементи са симетрични спрямо главния диагонал: а10 и а01; а20 и а02; а21 и а12; а30 и а03 и т.н.

а00 а01 а02 а03

а10 а11 а12 а13

а20 а21 а22 а23

а30 а31 а32 а33

VII.Рекурсия

Задача 22

Програма за изчисляване на факториел чрез рекурсивна функция.

#include "stdafx.h"

#include <iostream>

using namespace std;

int fact(int n)

{

if (n <= 1)

return 1;

else

return n\*fact(n-1);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int n;

cout << "n = ";

cin >> n;

cout << "n! = " << fact(n) << "\n";

return 0;

}

Задача 23

Програма за изчисляване на факториел чрез итерация.

#include "stdafx.h"

#include <iostream>

using namespace std;

int fact(int n)

{

if(n == 0)

return 1;

int i, f = 1;

for(i=1; i<=n; i++)

f = f\*i;

return f;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int n;

cout << "n = ";

cin >> n;

cout << "n! = " << fact(n) << "\n";

return 0;

}

Задача 24\*

Съставете алгоритъм и напишете програма за преобразуване на цяло десетично положително число в двоично чрез рекурсивна функция.

Задача 25\*

Съставете алгоритъм и напишете програма за изчисляване на xn (n - цяло число) чрез рекурсивна функция, съгласно следните формули:

xn = x\*xn-1, n>0

xn = 1, n=0

xn = 1/x-n, n<0.

Задача 26

Програма за изчисляване на най-големия общ делител на две естествени числа чрез рекурсивна функция.

#include "stdafx.h"

#include <complex>

#include <iostream>

using namespace std;

int min(int p, int q)

{

if (p > q)

return q;

else

return p;

}

int nod(int c, int d)

{

if( c == d)

return c;

else

return nod(abs(c-d), min(c,d));

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int a, b;

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

cout << "NOD = " << nod(a,b) << "\n";

return 0;

}

Задача 27\*

Съставете алгоритъм и напишете програма за изчисляване на сумата на елементите на даден едномерен масив чрез рекурсивна функция.

Задача 28\*

Съставете алгоритъм и напишете програма, която проверява за наличие на дадена цифра в дадено естествено число чрез рекурсивна функция.

Задача 29

Програма, извеждаща цифрите на дадено естествено число в обратен ред чрез рекурсивна функция.

#include "stdafx.h"

#include <iostream>

using namespace std;

void reverse(int n)

{

if(n < 10)

cout << n;

else

{

cout << n%10;

reverse(n/10);

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int n;

cout << "n: ";

cin >> n;

reverse(n);

cout << endl;

return 0;

}

Задача 30

Програма, проверяваща за наличие на елемент с дадена стойност, в даден едномерен масив чрез рекурсивна функция.

#include "stdafx.h"

#include <iostream>

using namespace std;

#define N 5

int a[N];

int search(int x, int n)

{

if(n<1)

return a[0]==x;

return x==a[n-1] || search(x,n-1);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int i, n;

for (i=0;i<N;i++)

{

cout << "array[" << i << "] = ";

cin >> a[i];

}

cout << "n: ";

cin >> n;

cout << "n = " << n;

if(search(n,i))

cout << " e ";

else

cout << " ne e ";

cout << "element ot masiva" << endl;

return 0;

}

Задача 31

Програма, проверяваща дали дадена редица е монотонно намаляваща, т.е. дали всеки елемент на редицата е по-голям или равен на следващият чрез рекурсивна функция.

#include "stdafx.h"

#include <iostream>

using namespace std;

#define N 5

int a[N];

int monoton(int n)

{

int b;

if(n==1)

return 1;

if(a[n-2]>=a[n-1])

b=1;

else b=0;

return b && monoton(n-1);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int i;

for (i=0;i<N;i++)

{

cout << "array[" << i << "] = ";

cin >> a[i];

}

cout<<"Redicata ";

if(monoton(i))

cout<<"e ";

else

cout<<"ne e ";

cout << "monotonno namalyvashta. " << endl;

return 0;

}

Задачи за изпълнение

1. Изпълнете, разгледайте и анализирайте решените задачи – номера 22, 23, 26, 29, 30 и 31.

2. Съставете алгоритъм и напишете програмен код на нерешените задачи – номера 21\*, 24\*, 25\*, 27\* и 28\*.