

Операторы цикла

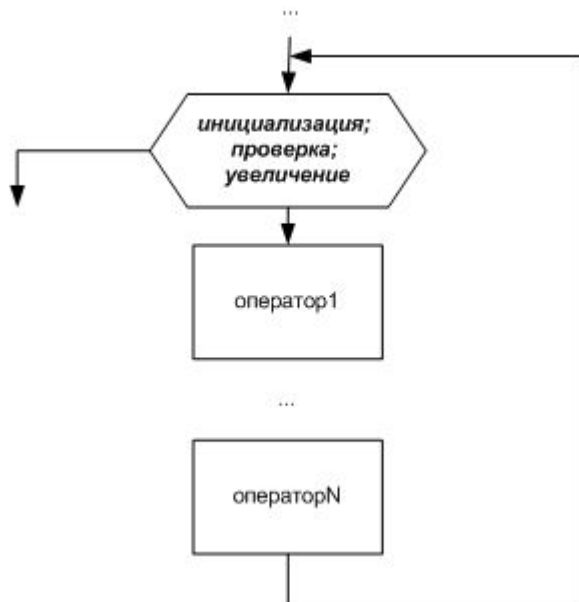
- Для повторения операторов определенное число раз ваши программы используют оператор C++ *for*.
- С помощью оператора C++ *while* программы повторяют операторы до тех пор, пока указанное условие истинно.
- Оператор C++ *do while* позволяет программам выполнять операторы по крайней мере один раз, а затем, возможно, повторять операторы, основываясь на определенном условии.

Пошаговый цикл

*for (инициализация; проверка; увеличение)
{ операторы;}*

**for (count = 1; count <= 10; count++)
оператор;**

for	<u>(count = 1;</u>	<u>count <= 10;</u>	<u>count++)</u>
	<i>Инициализация</i>	<i>Проверка</i>	<i>Увеличение</i>



Приклад. Скласти програму для розрахунку $n!$

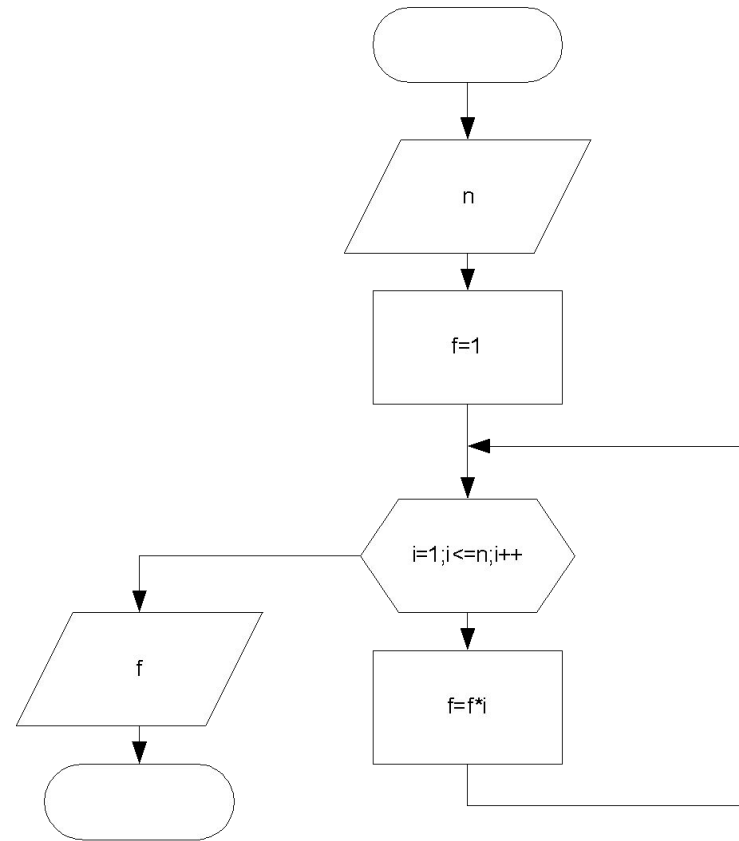
1. Постановка задачі

Вхід	Дія	Вихід
n - ціле	1. Введення n 2. $f=1$; 3. $i=1; i \leq n; i++$ 3.1 $f=f*i$ 4. Виведення f	f – довге ціле

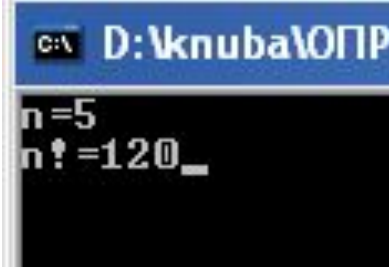
3. Тестовий приклад

$n=5$ $n!=120$
 $n=4$ $n!=24$
 $n=6$ $n!=720$

2. Алгоритм

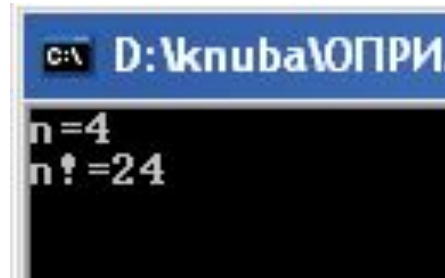


```
#include "stdafx.h"
#include "iostream"
#include "stdio.h"
#include "math.h"
#include "conio.h"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int n,i;
    long f=1;
    cout<<"n=";
    cin>>n;
    for(i=1;i<=n;i++)
        f=f*i;
    cout<<"n!="<<f;
    getch();
    getch();
    return 0;
}
```



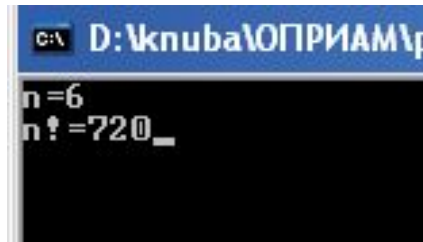
C:\ D:\knuba\OПP

```
n=5
n! =120_
```



C:\ D:\knuba\OПPPI

```
n=4
n! =24
```



C:\ D:\knuba\OПPPIAM\p

```
n=6
n! =720_
```

Приклад. Скласти програми мовою Сі++ згідно з варіантом

$$25. \quad y = 1 + \underbrace{\frac{2}{3^2} + \frac{3}{5^2} + \dots}_n$$

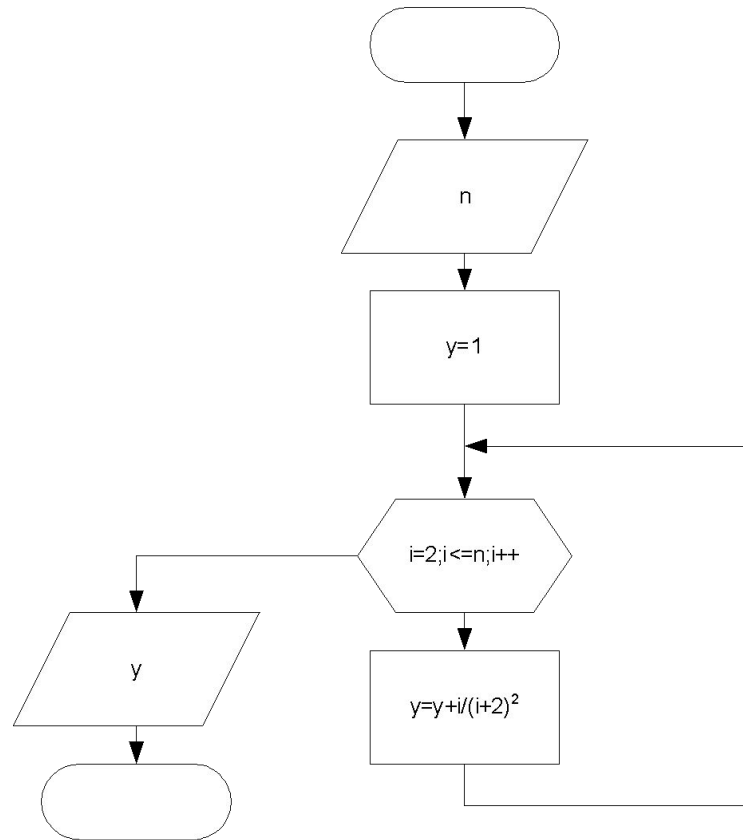
1. Постановка задачі

Вхід	Дія	Вихід
n - ціле	1. Введення n 2. y=1; 3. i=2; i<=n; i++ 3.1 y=y+i/(i+2) ² 4. Виведення y	y – дійсне

3. Тестовий приклад

n=3 y=1.4
n=2 y=1.22
n=4 y=1.56

2. Алгоритм



```
#include "stdafx.h"
```

```
#include "iostream"
```

```
#include "stdio.h"
```

```
#include "math.h"
```

```
#include "conio.h"
```

```
using namespace std;
```

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

```
{
```

```
int n;
```

```
double y=1;
```

```
cout<<"n=";
```

```
cin>>n;
```

```
for(int i=2;i<=n;i++)
```

```
y=y+i/pow(i+1,2.0);
```

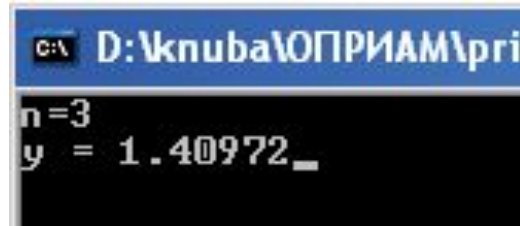
```
cout<<"y = "<<y;
```

```
getchar();
```

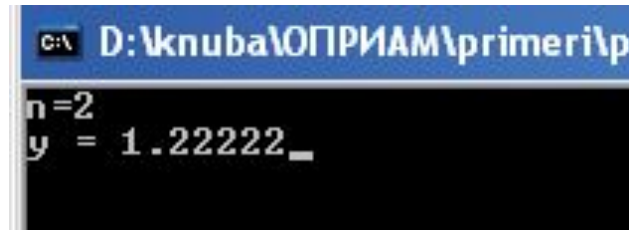
```
getchar();
```

```
return 0;
```

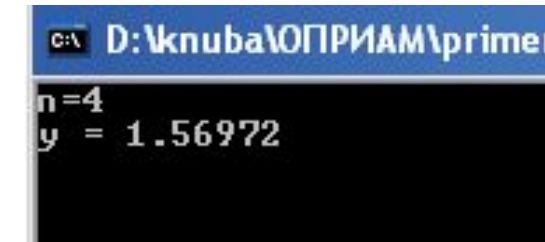
```
}
```



```
C:\ D:\knuba\ОПРИАМ\pri
n=3
y = 1.40972_
```



```
C:\ D:\knuba\ОПРИАМ\primer\p
n=2
y = 1.22222_
```

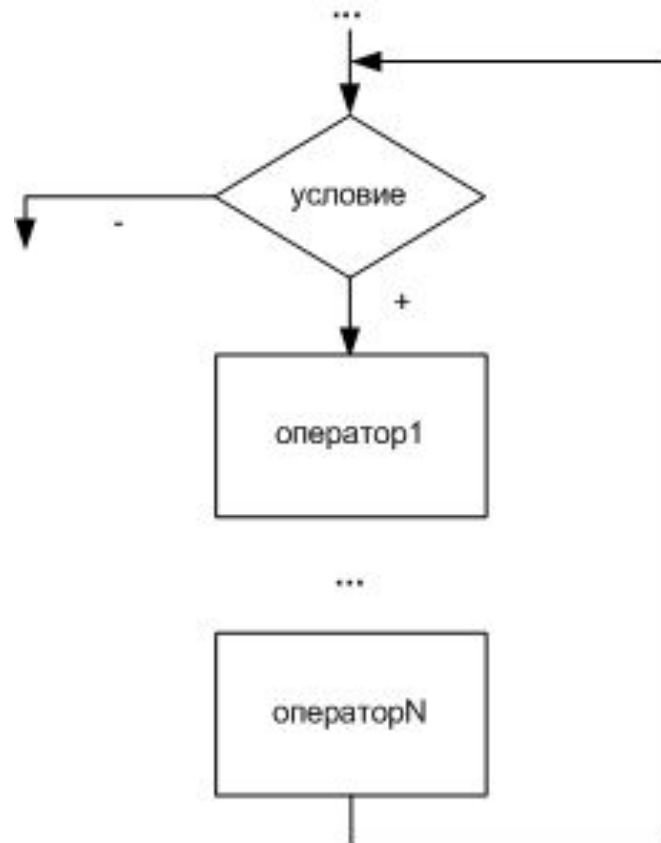


```
C:\ D:\knuba\ОПРИАМ\primer
n=4
y = 1.56972
```

Цикл с предусловием

*while (условие)
оператор;*

Сталкиваясь с оператором *while*, программа будет оценивать условие цикла. Если условие истинно, ваша программа выполняет операторы цикла *while*. После выполнения последнего оператора цикла программа снова проводит проверку условия. Если условие истинно, программа повторит этот процесс, выполнит операторы, а затем повторит проверку условия. Если условие оценивается как ложь, программа продолжит свое выполнение с первого оператора, который следует за оператором *while*.



Приклад. Скласти програму для розрахунку $n!$

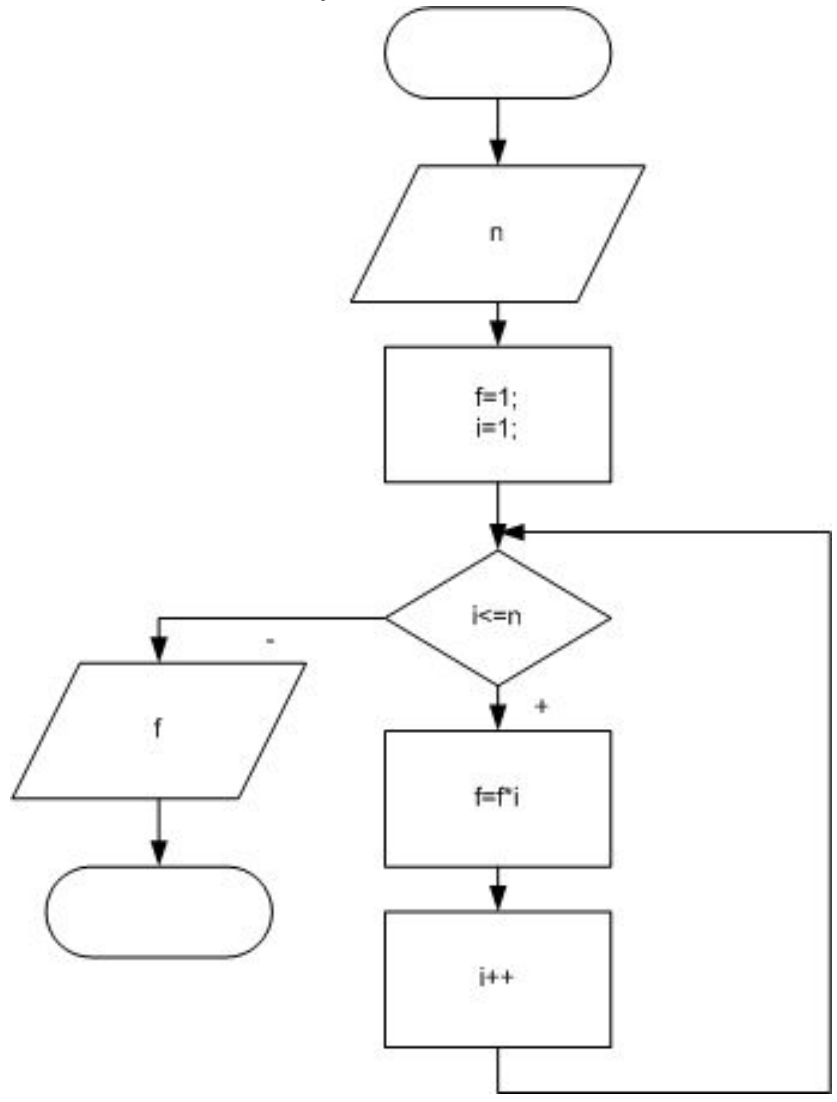
1. Постановка задачі

Вхід	Дія	Вихід
n - ціле	1. Введення n 2. $f=1; i=1;$ 3. Доки $i \leq n$ 3.1 $f=f*i$ 3.2 $i++$ 4. Виведення f	f - довге ціле

3. Тестовий приклад

$n=5 \quad n!=120$
 $n=4 \quad n!=24$
 $n=6 \quad n!=720$

2. Алгоритм



```

#include "stdafx.h"
#include "stdio.h"
#include "math.h"
#include "conio.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
int n,i=1;
long f=1;
cout<<"n=";
cin>>n;
while(i<=n)
{
f=f*i;
i++;
}
cout<<"n!="<<f;
getchar();
getchar();
return 0;
}

```

```

C:\ D:\knuba\OPIAM\primer\prim
n=4
n!=24

```

```

C:\ D:\knuba\OPIAM\primer\prim3_le
n=5
n!=120_

```

```

C:\ D:\knuba\OPIAM\primer\prim
n=6
n!=720_

```

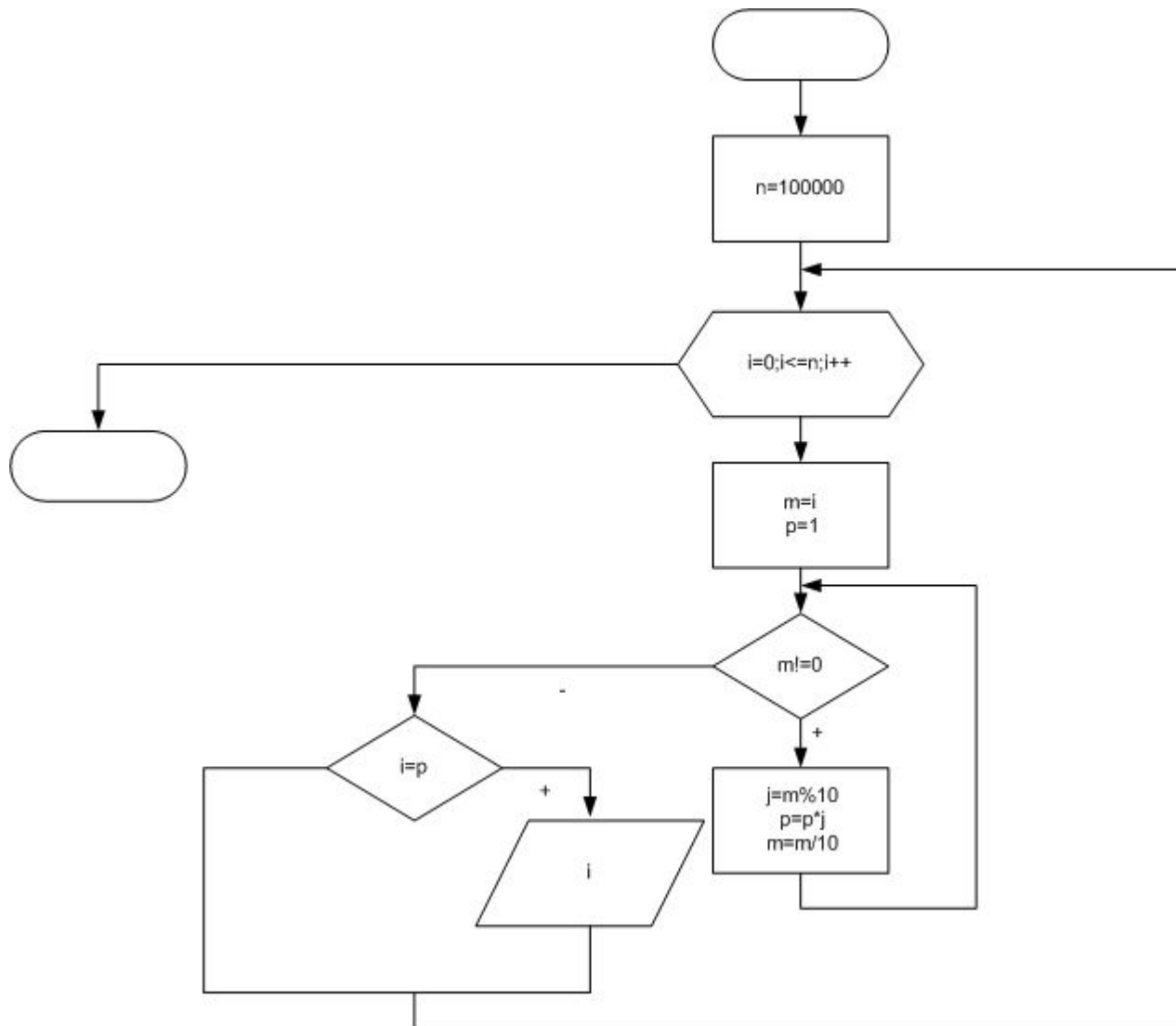
Приклад. Вивести на екран всі “досконалі” числа в діапазоні від 0 до 100000. Число “досконале”, якщо воно дорівнює добутку своїх цифр

1. Постановка задачі

Вхід	Дія	Вихід
	1. $n=100000$	i – довге
	2. $i=0; i \leq n; i++$	ціле
	2.1 $m=i$	
	2.2. $p=1$	
	2.3 Доки $m \neq 0$	
	2.3.1 $j=m \% 10$	
	2.3.2 $p=p*j$	
	2.3.3 $m=m/10$	
	2.4 Якщо $i=m$	
	тоді Вивід i	

2.Алгоритм

3. Тестовий приклад
0..9 – досконалі числа



```

#include "stdafx.h"
#include "stdio.h"
#include "math.h"
#include "conio.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
int n=100000;
long i,j,m,p=1;
for(i=0;i<=n;i++)
{
p=1;
m=i;
while (m!=0)
{
j=m%10;
p=p*j;
m=m/10;
}
if(i==p)
cout<<"Chislo "<<i<<" doskonale"<<endl;
}
getchar();
getchar();
return 0;
}

```

```

C:\ D:\knuba\ОПРИАМ\primeri\prim5
Chislo 1 doskonale
Chislo 2 doskonale
Chislo 3 doskonale
Chislo 4 doskonale
Chislo 5 doskonale
Chislo 6 doskonale
Chislo 7 doskonale
Chislo 8 doskonale
Chislo 9 doskonale

```

Приклад. Скласти програму для табулювання функції на заданому інтервалі і обчисленні

Певного значення.

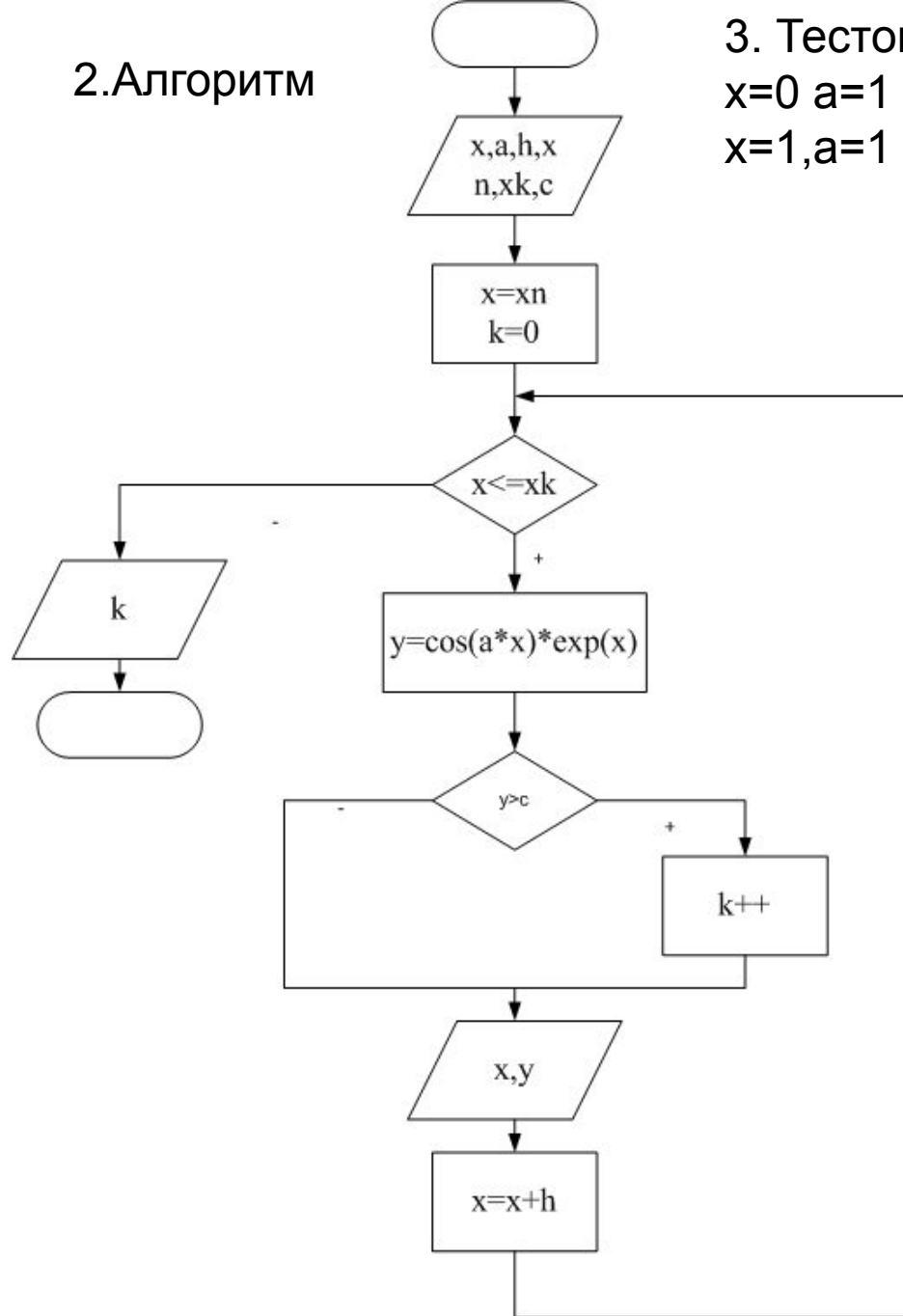
25. $y = \cos ax \cdot e^x$ при $x \in [0, 5]$

кількість значень y , що більші за задане C .

1. Постановка задачі

Вхід	Дія	Вихід
a, h, x_n, x_k, c - дійсні	<ol style="list-style-type: none"> 1. введення початкових значень 2. $x = x_n$ 3. $k = 0$ 4. Доки $x \leq x_k$ <ol style="list-style-type: none"> 4.1 $y = \cos(a \cdot x) \cdot \exp(x)$ 4.2 Якщо $y > c$ тоді $k++$ 4.3 Виведення x, y 4.4 $x = x + h$ 5. Виведення k 	<p>y-дійсне</p> <p>k - ціле</p>

2. Алгоритм



3. Тестовый пример
 $x=0$ $a=1$ $y = 2.72$
 $x=1, a=1$ $y=1.47$

с:\ D:\кнуба\ЮПРИАМ\primeri\prim6

xn=0	
xk=5	
h=0.1	
a=1	
c=1	
0.0000	1.0000
0.1000	1.0996
0.2000	1.1971
0.3000	1.2896
0.4000	1.3741
0.5000	1.4469
0.6000	1.5039
0.7000	1.5402
0.8000	1.5505
0.9000	1.5289
1.0000	1.4687
1.1000	1.3627
1.2000	1.2031
1.3000	0.9815
1.4000	0.6893
1.5000	0.3170
1.6000	-0.1446
1.7000	-0.7053
1.8000	-1.3745
1.9000	-2.1615
2.0000	-3.0749
2.1000	-4.1227
2.2000	-5.3112
2.3000	-6.6456
2.4000	-8.1284
2.5000	-9.7599
2.6000	-11.5369
2.7000	-13.4524
2.8000	-15.4945
2.9000	-17.6463
3.0000	-19.8845
3.1000	-22.1788
3.2000	-24.4907
3.3000	-26.7732
3.4000	-28.9692
3.5000	-31.0112
3.6000	-32.8198
3.7000	-34.3034
3.8000	-35.3572
3.9000	-35.8628
4.0000	-35.6877
4.1000	-34.6850
4.2000	-32.6937
4.3000	-29.5388
4.4000	-25.0325
4.5000	-18.9752
4.6000	-11.1574
4.7000	-1.3621
4.8000	10.6320
4.9000	25.0467
5.0000	42.0992

k = 15

```

#include "stdafx.h"
#include "stdafx.h"
#include "stdio.h"
#include "math.h"
#include "conio.h"
#include <iostream>
#include <iomanip>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
double x,xn,xk,h,c,y,a;
int k=0;
cout<<"xn=";
cin>>xn;
cout<<"xk=";
cin>>xk;
cout<<"h=";
cin>>h;
cout<<"a=";
cin>>a;
cout<<"c=";
cin>>c;
x=xn;

```

```

while (x<=xk)
{
y=cos(a*x)*exp(x);
if (y>c) k++;
cout<< fixed << setprecision(4)<<" | "<<x<<" |
"<<y<<" |"<<endl;
x=x+h;
}
cout<<"k = "<<k;
getchar();
getchar();
return 0;
}

```

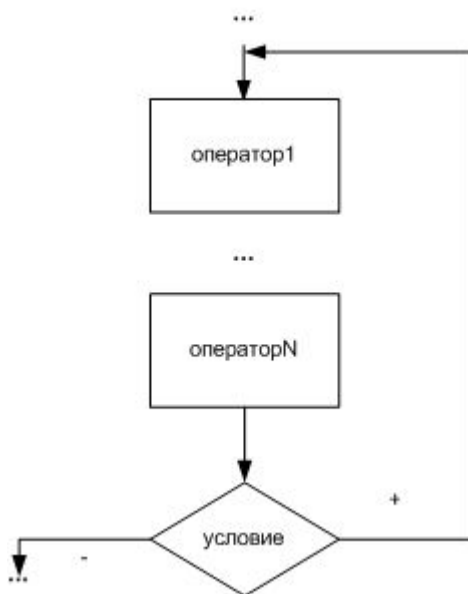

Цикл постусловием

do {

оператор;

} while (условие);

- **Когда программа встречает оператор *do while*, она сразу же выполняет операторы, содержащиеся в цикле. Затем программа исследует условие цикла. Если условие истинно, программа повторяет операторы цикла и процесс продолжается. Если условие цикла становится ложным, программа продолжает свое выполнение с первого оператора, за оператором *do while*.**



Приклад. Скласти програму для розрахунку n!

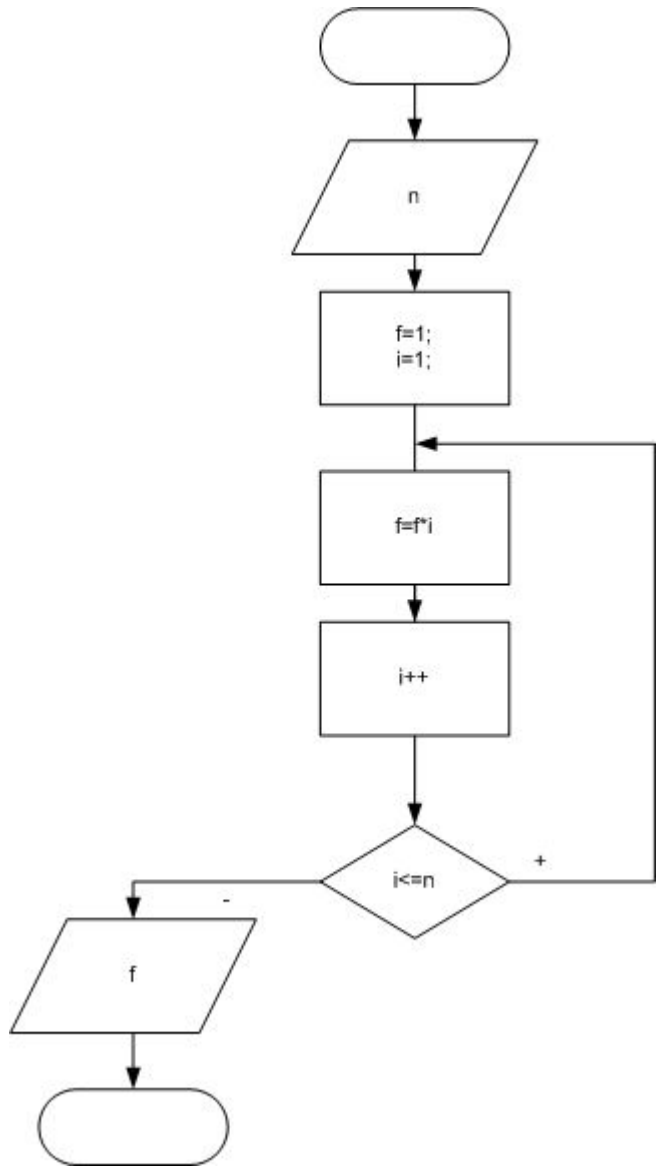
1. Постановка задачі

Вхід	Дія	Вихід
n - ціле	1. Введення n 2. f=1; i=1; 3. 3.1 f=f*i 3.2 i++ 3.3 Якщо i<=n тоді повернення на крок 3.1 Інакше 4. Виведення f	f – довге ціле

3. Тестовий приклад

n=5 n!=120
n=4 n!=24
n=6 n!=720

2.Алгоритм



```

#include "stdafx.h"
#include "stdio.h"
#include "math.h"
#include "conio.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
int n,i=1;
long f=1;
cout<<"n=";
cin>>n;
do
{
f=f*i;
i++;
}
while(i<=n);
cout<<"n!="<<f;
getchar();
getchar();
return 0;
}

```

```

C:\> D:\knyba\OPIAM\primeri\prim
n=4
n!=24

```

```

C:\> D:\knyba\OPIAM\primeri\prim3_le
n=5
n!=120_

```

```

C:\> D:\knyba\OPIAM\primeri\
n=6
n!=720_

```