Изчислителен процес. Блок схеми. Състояние на програмата. Блокове в C++. Начални сведения за масиви и символни низове

Калин Георгиев

27 октомври 2015 г.

Препоръчителна литература

Основна

- М. Тодорова. Програмиране на C++ - първа част, С. СИЕЛА , 2002.
- М. Тодорова, П. Армянов, Д. Зотева, К. Георгиев. Сборник от задачи по програмиране на C++. Част първа. Увод в програмирането, ТехноЛогика ЕООД, 2008.

Допълнителна

- Niklaus Wirth, Algorithms + Data Structures = Programs, Prentice-Hall Series in Automatic Computation
- Robert Sedgewick, Algorithms in C++
- П. Наков, П. Добриков, Програмиране = ++Алгоритми; С., Top Team Co, 2003. Допълнителна:
- Л. Амерал, Алгоритми и структури от данни в C++, С, СОФТЕХ, 2001.
- Липман, Езикът C++ в примери, С., КОЛХИДА ТРЕИД – КООП, 1993.

Изчислителен процес

Последоватленост на операциите и “състояние”

```
int a,b;

a = 5;

b = 10;

b = a + b;

b = a + b;
```

...	a	b	...
...	?	?	...

...	a	b	...
...	5	?	...

...	a	b	...
...	5	10	...

...	a	b	...
...	5	15	...

...	a	b	...
...	5	20	...

Последоватленост на операциите и "състояние"

```
int a,b;

a = 5;

b = 10;

b = a + b;

b = a + b;
```

...	a	b	...
...	?	?	...

...	a	b	...
...	5	?	...

...	a	b	...
...	5	10	...

...	a	b	...
...	5	15	...

...	a	b	...
...	5	20	...

Последоватленост на операциите и “състояние”

```
int a,b;

a = 5;

b = 10;

b = a + b;

b = a + b;
```

...	a	b	...
...	?	?	...

...	a	b	...
...	5	?	...

...	a	b	...
...	5	10	...

...	a	b	...
...	5	15	...

...	a	b	...
...	5	20	...

Последоватленост на операциите и “състояние”

```
int a,b;

a = 5;

b = 10;

b = a + b;

b = a + b;
```

...	a	b	...
...	?	?	...

...	a	b	...
...	5	?	...

...	a	b	...
...	5	10	...

...	a	b	...
...	5	15	...

...	a	b	...
...	5	20	...

Последоватленост на операциите и “състояние”

```
int a,b;

a = 5;

b = 10;

b = a + b;

b = a + b;
```

...	a	b	...
...	?	?	...

...	a	b	...
...	5	?	...

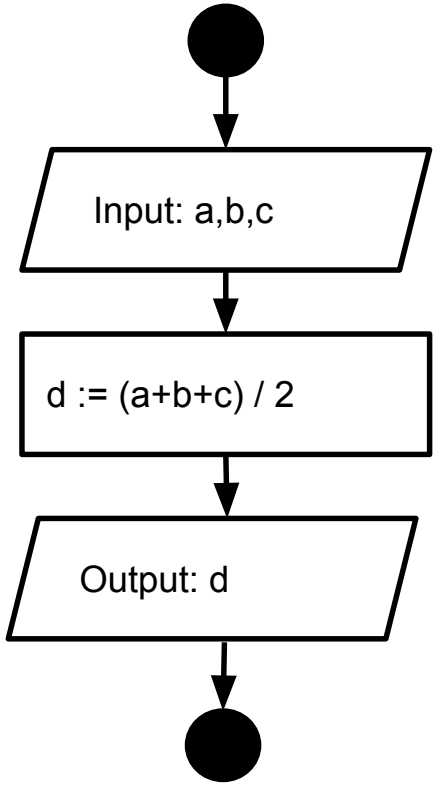
...	a	b	...
...	5	10	...

...	a	b	...
...	5	15	...

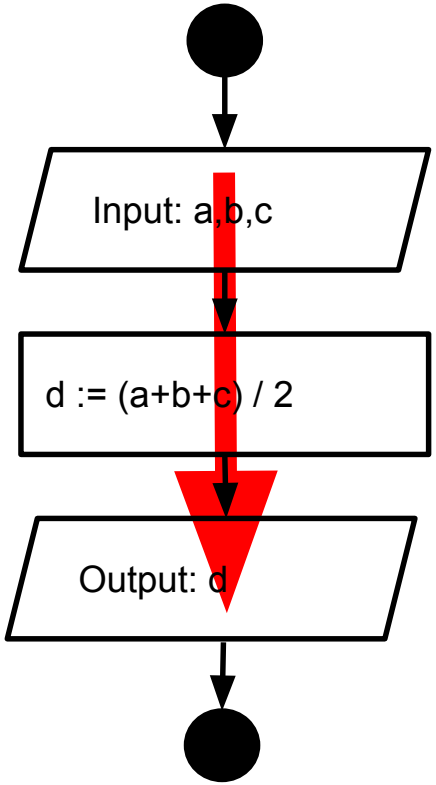
...	a	b	...
...	5	20	...

Блок схеми и процеси

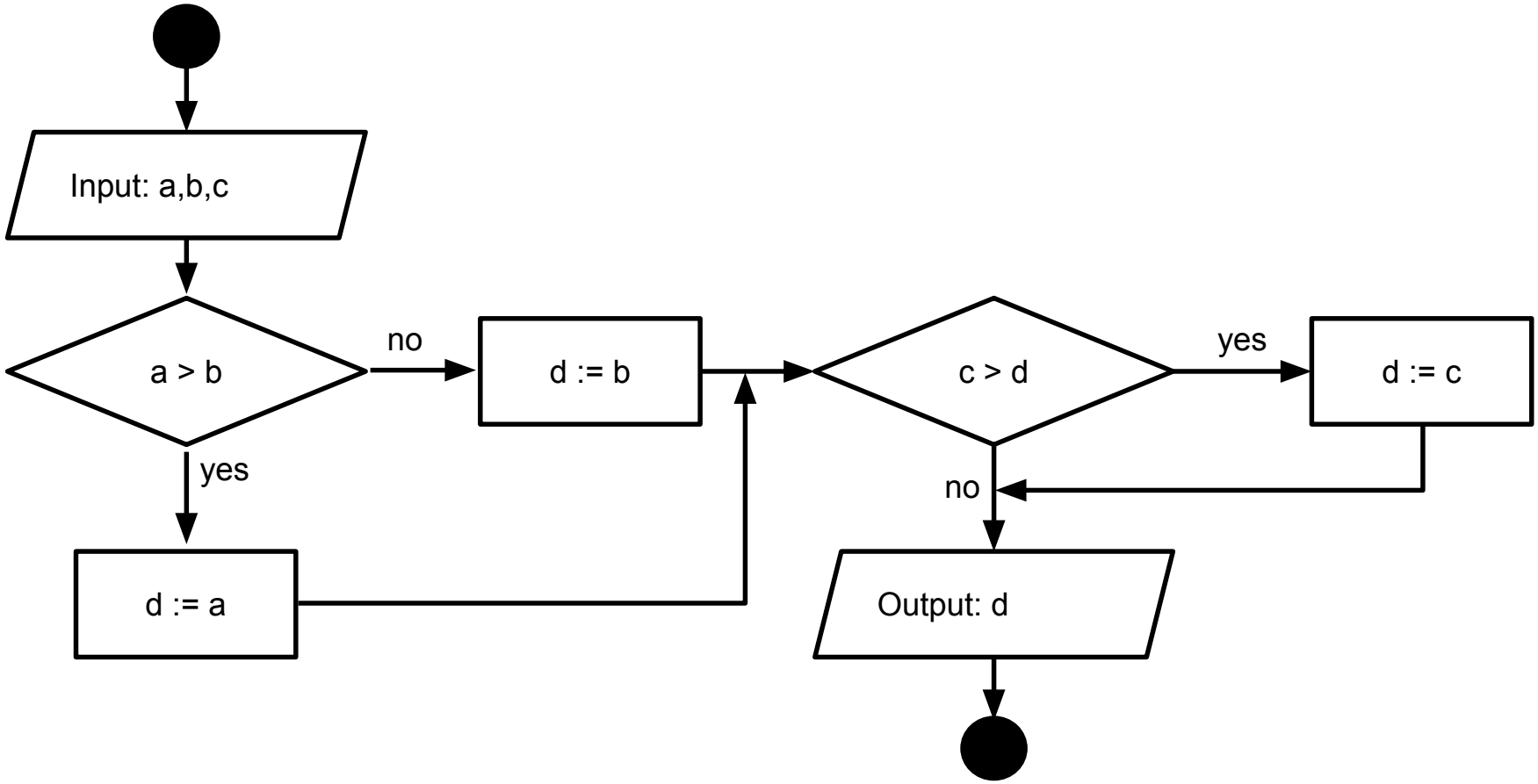
Линейна програма



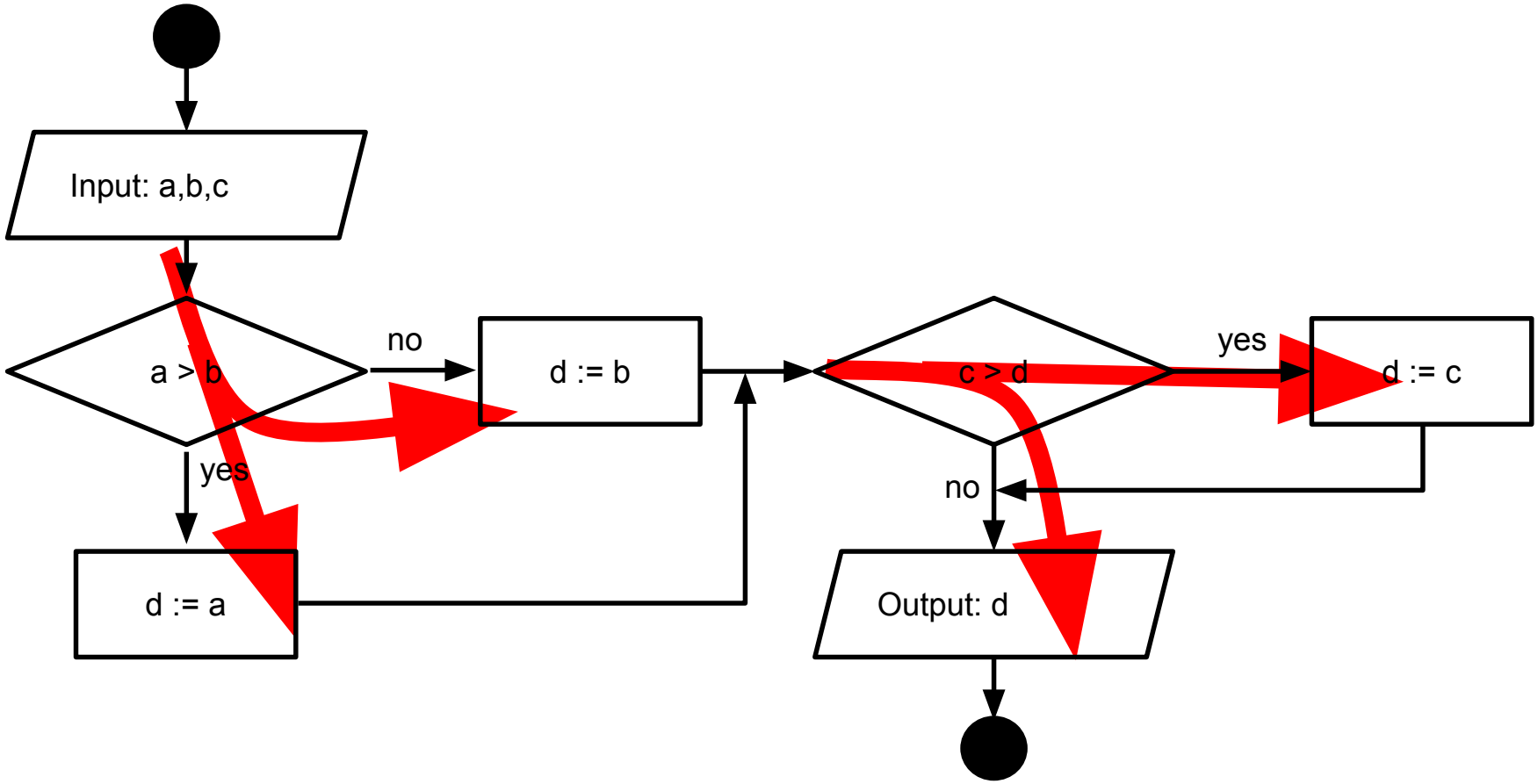
Линейна програма



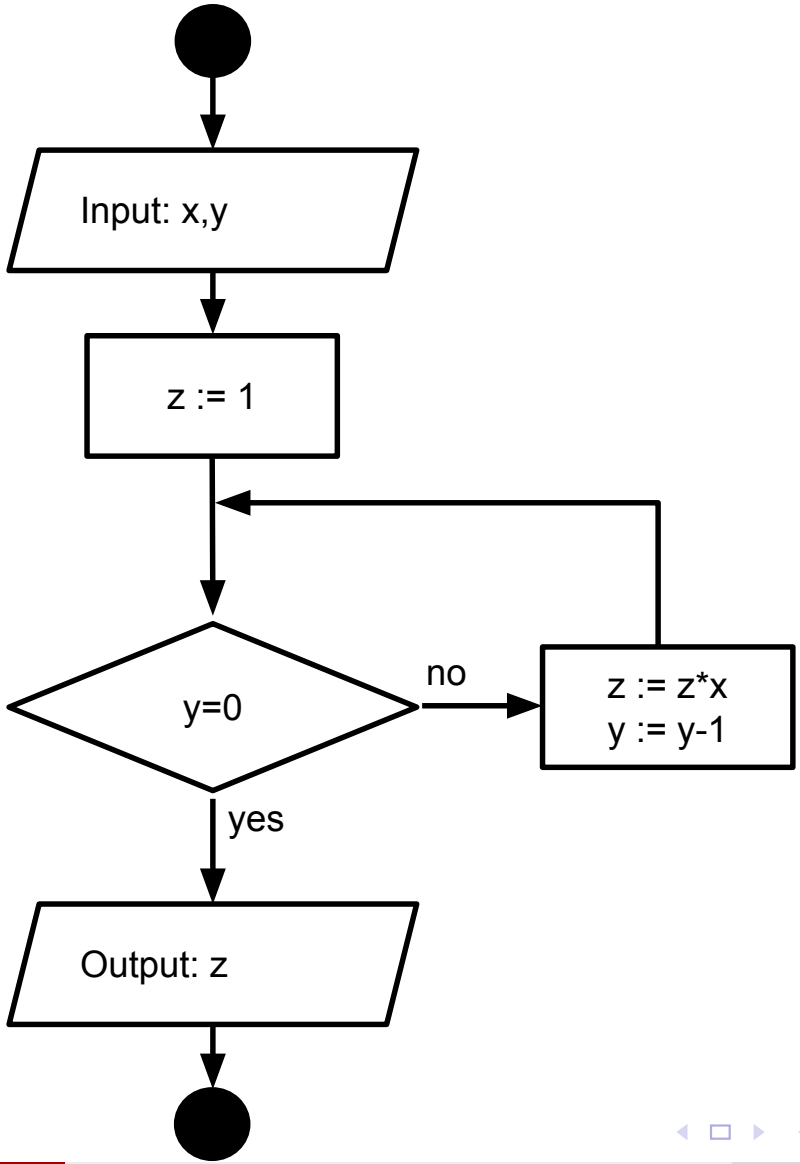
Разклонение



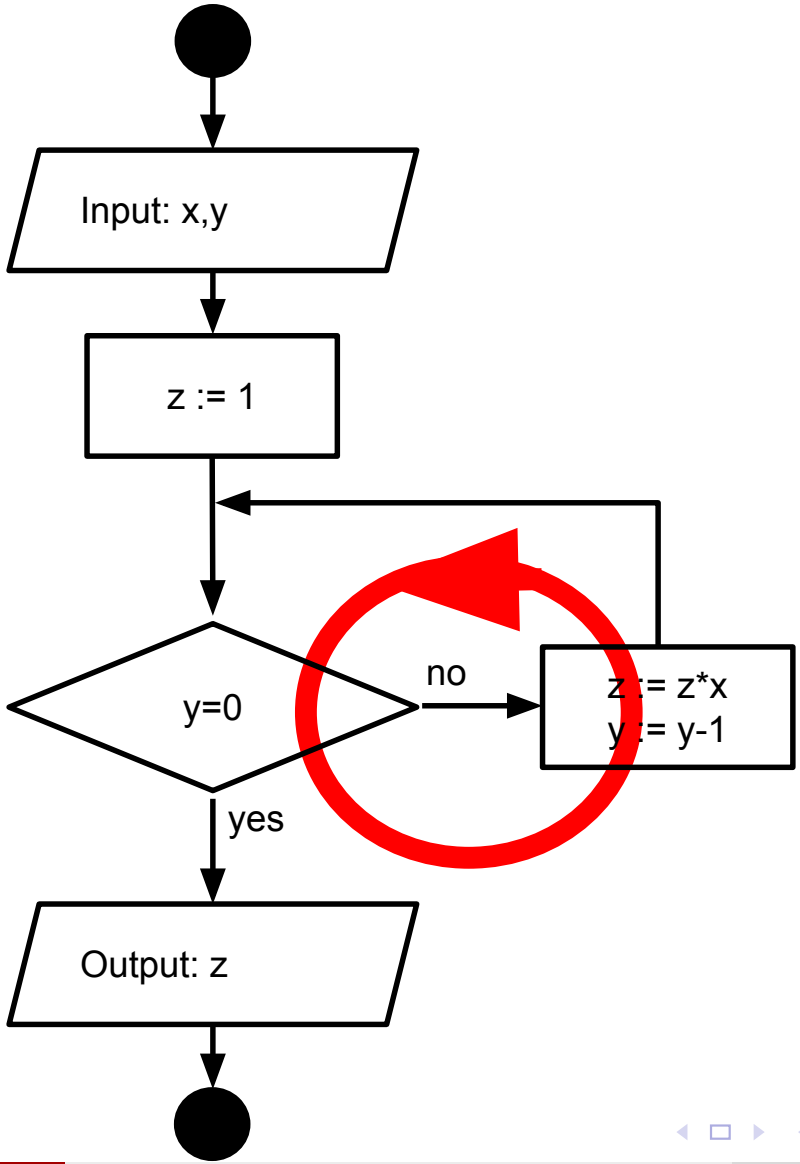
Разклонение



Цикъл

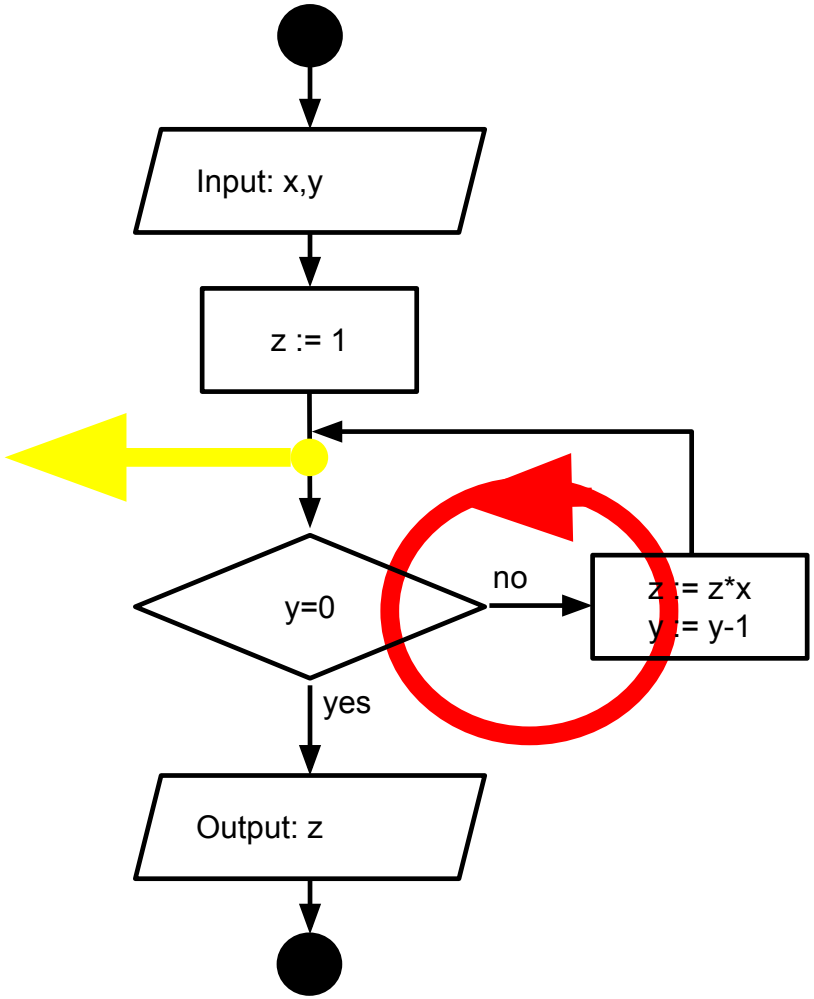


Цикъл



x^y

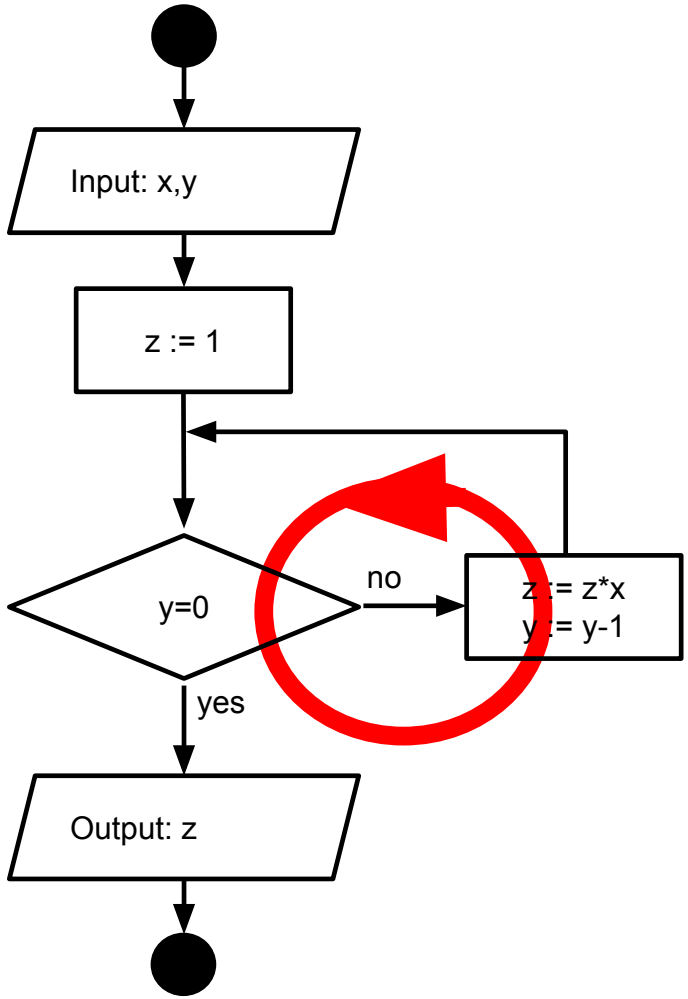
#	x	y	z
0	2	5	1
1	2	4	2
2	2	3	4
3	2	2	8
4	2	1	16
5	2	0	32



Още един вид цикъл в C++

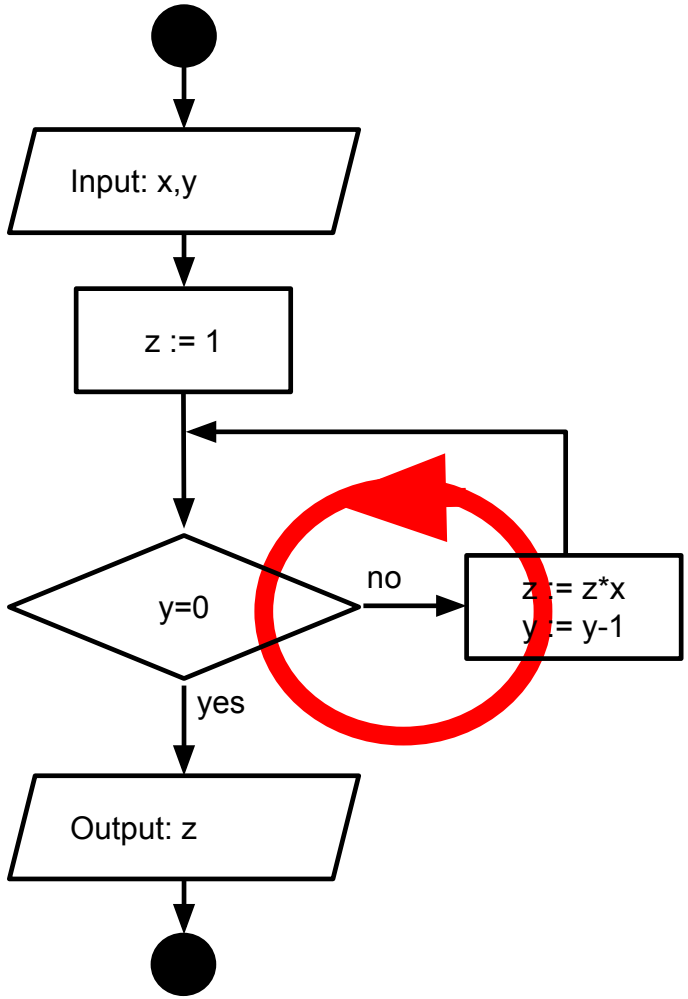
Неструктурирани езици

```
10. Въведи X и Y
20. Z := 1
30. Ако Y == 0 GOTO 70
40. Z := Z * X
50. Y := Y -1
60. GOTO 30
70. Отпечатай Z
80. Край
```



Цикъл While

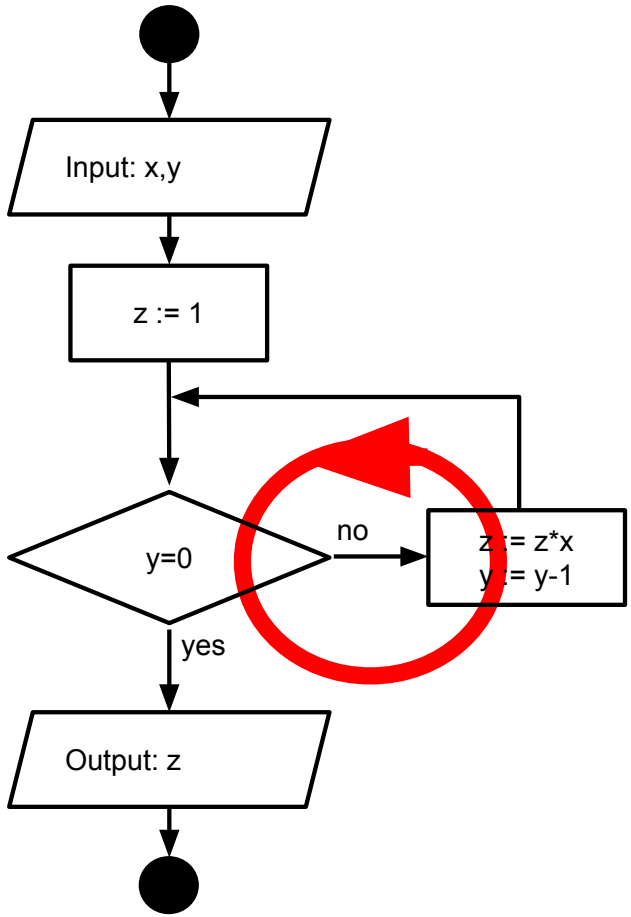
```
int x,y,z;

cin >> x >> y;

z = 1;

while (y != 0)
{
  z = z * x;
  y--;
}

cout << z;
```



Сравнение

```
int x,y,z;

cin >> x >> y;

z = 1;

while (y != 0)
{
   z = z * x;
   y--;
}

cout << z;
```

```
10. Въведи X и Y
20. Z := 1
30. Ако Y == 0 GOTO 70
40. Z := Z * X
50. Y := Y -1
60. GOTO 30
70. Отпечатай Z
80. Край
```



Примери

- Намиране на броя на цифрите в десетичния запис на естествено число
- $n!$
- Намиране на сумата на цифрите в десетичния запис на естествено число
- Проверка дали дадено естествено число притежава цифрата 5 в десетичния си запис

Блокове в C++

Какво е блок?

```
int a = 0;

int b = a + 10;

if (a == 0)
{
   int b = a;
   cout << b;
   b = 100;
}

cout << b;
```

Стек от променливи

```
int a = 0;
int b = a + 10;
```

...	a	b	...
...	0	10	...

```
if (a == 0)
{
   int b = a;
   cout << b;
```

...	a	b	b	...
...	0	10	0	...

```
   b = 100;
```

...	a	b	b	...
...	5	10	100	...

```
}
```

...	a	b	X	...
...	0	10	X	...

```
cout << b;
```

...	a	b	...
...	0	10	...

Стек от променливи

```
int a = 0;
int b = a + 10;
```

...	a	b	...
...	0	10	...

```
if (a == 0)
{
   int b = a;
   cout << b;
```

...	a	b	b	...
...	0	10	0	...

```
   b = 100;
```

...	a	b	b	...
...	5	10	100	...

```
}
```

...	a	b	X	...
...	0	10	X	...

```
cout << b;
```

...	a	b	...
...	0	10	...

Стек от променливи

```
int a = 0;
int b = a + 10;
```

...	a	b	...
...	0	10	...

```
if (a == 0)
{
   int b = a;
   cout << b;
```

...	a	b	b	...
...	0	10	0	...

```
   b = 100;
```

...	a	b	b	...
...	5	10	100	...

```
}
```

...	a	b	X	...
...	0	10	X	...

```
cout << b;
```

...	a	b	...
...	0	10	...

Стек от променливи

```
int a = 0;
int b = a + 10;
```

...	a	b	...
...	0	10	...

```
if (a == 0)
{
   int b = a;
   cout << b;
```

...	a	b	b	...
...	0	10	0	...

```
   b = 100;
```

...	a	b	b	...
...	5	10	100	...

```
}
```

...	a	b	X	...
...	0	10	X	...

```
cout << b;
```

...	a	b	...
...	0	10	...

Стек от променливи

```
int a = 0;
int b = a + 10;
```

...	a	b	...
...	0	10	...

```
if (a == 0)
{
   int b = a;
   cout << b;
```

...	a	b	b	...
...	0	10	0	...

```
   b = 100;
```

...	a	b	b	...
...	5	10	100	...

```
}
```

...	a	b	X	...
...	0	10	X	...

```
cout << b;
```

...	a	b	...
...	0	10	...

Първоначални сведения за масиви и низове

Представяне на низове

- ASCii таблица

...	A^{65}	B^{66}	C^{67}	...

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Представяне на низове

- ASCii таблица

...	A^{65}	B^{66}	C^{67}	...

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Представяне на низове

- ASCii таблица

...	A^{65}	B^{66}	C^{67}	...

- Представяне в паметта

...	H	E	L	L	O	_	...
...	72	69	76	76	79	0	...

- Масиви

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];
cin >> arr[18];
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)
{
   cout << arr[count];
}
```

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];
cin >> arr[18];
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)
{
   cout << arr[count];
}
```

Масиви

- Дефиниране чрез тип и размер:

```
int arr[100];
```

- Достъп до всеки отделен елемент:

```
int b = arr[18];
cin >> arr[18];
b = arr[1] + arr[2];
```

- Обхождане с for цикъл

```
for (int count = 0; count < 100; count++)
{
   cout << arr[count];
}
```

Прост пример с низове

```
int main ()
{

  char str[100] = "Hello world!";
  cout << str << endl;

  str[0] = 'Y';
  cout << str << endl;

  cout << "Please input a string:";
  cin >> str;

  for (int counter = 0; counter < 100; counter++)
  {
    if (str[counter] == 'a')
    {
      str[counter] = 'b';
    }
  }

  cout << str << endl;

  return 0;
}
```

Какво не можем да праивм с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване (a=b)
- Сравнение (a==b, a < b,...)

Какво не можем да праивм с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване (a=b)
- Сравнение (a==b, a < b,...)

Какво не можем да праивм с масиви и низове

- Няма проверка за коректност!

```
char a[6] = "HELLO";
a[10] = '!';
```

...	0	1	2	3	4	5	...	10	...
...	H	E	L	L	O	_	...	!	...

- Присвояване (a=b)
- Сравнение (a==b, a < b,...)

Вградени функции за работа с низове

Дължина на низ

```
cout << strlen(a);
```

Присвояване на низове

```
strcpy (d,a);
strcpy (c,a); //!!!
```

Сравнение на низове

```
if (strcmp (a,b) < 0)
  {cout << "a<b";}
else
  {cout << "b<=a";};
```

Конкатенация на низове

```
strcpy (d,a); //d -> "HELLO"
```

...	0	1	2	3	4	5	6	7	8	9	10	...
...	H	E	L	L	O	_	?	?	?	?	?	...

```
strcat (d,b); //d -> "HELLOWORLD"
```

...	0	1	2	3	4	5	6	7	8	9	10	...
...	H	E	L	L	O	W	O	R	L	D	_	...

```
#include <cstring>
...
char a[6] = "HELLO";
char b[6] = "WORLD";
char c[4] = "BYE";
char d[11] = "???";
```

...	0	1	2	3	4	5	...
...	H	E	L	L	O	_	...

...	0	1	2	3	4	5	...
...	W	O	R	L	D	_	...

...	0	1	2	3	...
...	B	Y	E	_	...