Оператори за разклонения и цикли. Разработване на програми с разклонения и циклична структура. Съставяне и настройка на програми с основните видове цикли върху потокови данни. Оператори за цикъл.

Оператор if

В повечето програмни езици присъства възможност за разклоняване логиката на програмата – възможността да се изпълняват дадена част от кода само при изпълнено дадено условие, ако това условие не е изпълнено, кодът не се изпълнява. В програмният език C тази възможност се реализира чрез оператора if. Употребата се извършва по начин, показан по-долу:

```
if(<Условие>)
{
	// Код за изпълнение, ако условието е изпълнено
}
```

Под ***<условие>*** се разбира логическо условие, което може да бъде „истина“ или „лъжа“.

Оператор if-else

Употребата на този оператор е идентична на оператор if, но тук е предвиден код, който може да се изпълни само, когато условието не е изпълнено.

```
if(<Условие>)
{
	// Код за изпълнение, ако условието е изпълнено
}
else
{
	// Код за изпълнение, ако условието не е изпълнено
}
```

По този начин програмата има различно поведение в зависимост от истинността на поставеното условие.

Оператор switch-case

Идеята на този оператор е подобна на if, но тук има повече от две възможни стойности за тестване, като за всяка възможна стойност (случай) се изпълняват различни част от кода.

Употреба:

```
switch(<променлива>)
{
	case <стойност 1>:
		// Код за изпълнение 1
	case <стойност 2>:
		// Код за изпълнение 2
		break;
	case <стойност 3>:
		// Код за изпълнение 3
		break;
	default:
		// Код за изпълнение 4
		break;
}
```

Значение на елементите:

<променлива> - тук се записва името на променливата, чиято стойност ще бъде изследвана. Типът на променлива трябва да бъде задължително изброим тип (например цяло число).

<стойност1>, <стойност2>… – константи, които показват случаите, за които ще се изследва променливата.

case – запазена дума, която показва от къде започва изпълнението на кода, който се отнася за определената стойност. Изпълнението продължава до срещане на запазената дума break, възможно е break да се намира в следващ случай (както е показано в примера), така ако променливата има стойност 1, то ще се изпълни „Код за изпълнение 1“ и „Код за изпълнение 2“.

default – в този случай се влиза, ако стойността на променливата не е описана в предходните случаи.

Задачи за реализация:

1. Да се прочете от клавиатурата едно целочислено число и да се изведе съобщение, ако то е по-голямо от 6.

2. Да се прочете от клавиатурата едно целочислено число и да се изведе на екрана съобщение, ако остатъкът от делението му с 8 е по-голям от 4.
3. Да се прочетат от клавиатурата 3 числа и да се изведе на екрана най-малкото от тях.
4. Да се прочете от клавиатурата цифра и да се изведе на екрана нейното име. Да се използва оператор switch-case.
5. Да се прочете от клавиатурата едно число от 1 до 7 и да се изведе на екрана кой ден от седмицата съответства на това число. Да се предвиди случай, когато не е въведено валидно число.

Понятие за цикъл

Цикъл в терминологията на програмните езици представлява код, който може да се изпълнява няколко пъти последователно. Циклите се отличават от останалия код с това, че имат 4 важни елемента:

· инициализиране на брояч – използва се най- често, когато броят стъпки е краен и известен. В този елемент се задава началната стойност, за която да се извършва цикъла.

· условие за изпълнение – всеки цикъл задължително разполага с условие. В зависимост от истинността на това условие се взима решение дали цикъла да се изпълни още веднъж.

· актуализация на брояча – тук се задава следващата стойност на брояча, при който трябва да се изпълнява цикъла

· тяло – това е код, който ще се изпълнява в цикъла. В зависимост от вида си съществуват два основно вида цикли:

- цикли с предусловие – цикъл, за който преди изпълнението на тялото се проверява истинността на зададеното условие

- цикли с пост условие – цикъл, за който проверката за истинността на зададеното условие се проверява след изпълнението на тялото. Циклите от този тип се изпълняват поне веднъж, независимо от това дали е изпълнено или не условието.

В програмният език C съществуват 3 вида цикли – два от тях са с предусловие и един с пост условие – ***for, while, do-while***. И при трите цикъла е валидно правилото *„докато условието е вярно, до тогава ще се изпълнява тялото“*.

Единично изпълнение на тялото на цикъл се нарича ***итерация***.

В програмирането се използва понятието „вложени цикли“, което означава, че в тялото на даден цикъл може да има друг цикъл. Вложените цикли се използват за обхождане на данни като таблици, по-сложни списъци и други. Пример за използването им в таблици - единият цикъл обхожда редовете, другият колоните. Комбинацията от номер на ред и номер на колона прави достъпен съответния елемент в таблицата.

Цикъл FOR

```
for(<инициализация>; <условие>; <актуализация>)
{
	// Тяло
}
```

В този вид цикъл първоначалната инициализацията е част от синтаксиса, както и актуализацията. Употребата на цикъл for е главно, когато е известно колко точно стъпки трябва да се изпълни цикъла.

Примерна задача

Да се напише програмен фрагмент, който да отпечата числата от 1 до 10 на екрана.

```
int i;
...
for(i = 1; i<=10; i=i+1)
{
	printf("%d ",i);
}
```

В показаното решение ясно личат как данните от условието на задачата се прилагат в решението.

Изпълнението на програмния фрагмент е следното:

1. Променливата i приема стойност 1;

2. Проверява се истинността на условието i<=10 (1<=10). В този случай условието е „истина“, следователно се преминава към изпълнение на тялото;

3. Отпечатва се текущата стойност на i на екрана;

4. Край на тялото;

5. Преминава се към актуализация – в дадения пример към стойността на i се прибавя единица. Следователно новата стойност на i е 2;

6. Проверява се истинността на условието за новата стойност i<=10 (2<=10). Преминава се към изпълнение на тялото;

7. Изпълнение на тялото – отпечатване на стойността на i на екрана;

8. Актуализация на i;

... Пропускане на няколко стъпки от изпълнението на програмния фрагмент…

9. След извършване на актуализация i има стойност 11.

10. При проверката на условието i<=10 (11<=10) се установява, че условието не е изпълнено, следователно се напуска изпълнението на цикъла и се изпълнява програмния код след това. След изпълнението на цикъла стойността на i е 11. Това е добре да се знае в случаите, когато се налага употребата на същата променлива

Безкраен цикъл

Цикъл for може да бъде настроен да работи безкрайно. Това се постига по следния начин:

```
for(;;)
{
	//Тяло
}
```

Важно: Когато се използва безкраен цикъл трябва да се предвидят условия за неговото насилствено прекратяване.

Повече от една актуализация и една инициализация

В програмният език C е допустимо да се извършат повече от една инициализация и повече от една актуализация в цикъл for, като на мястото за инициализация се опишат всички начални стойности на променливите, разделение със запетая. По същия начин актуализирането на променливите се осъществява чрез изреждане на всички променливи, които трябва да се актуализират.

Пример:

```
int i,j;
for(i = 0, j = 0; i*j <= 100; i=i+1, j = j + 2)
{
	//Тяло
}
```

Важно: Не е задължително актуализацията или началната стойност на всички променливи да се записват на тези места, те могат да бъдат написани и като изпълними редове.

Цикъл WHILE

Начин на употреба

```
<инициализация>;
while(<условие>)
{
	// Тяло
	<актуализация>;
}
```

В този вид цикъл, както се вижда от синтаксиса, инициализацията и актуализацията не са вградени в синтаксиса. Това дава свободата за разположението на тези елементи по преценка на автора на програмния код. Присъствието на тези елементи е препоръчително. Цикъл while е отново цикъл с предусловие – условието ще се провери преди изпълнението на тялото

Примерна задача

Да се напише програмен фрагмент, който да отпечата числата от 1 до 10 на екрана.

```
int i;
i = 1;
while( i <= 10 )
{
	printf("%d ", i);
	i = i + 1;
}
```

Стъпките на изпълнение тук са идентични, както при реализацията с цикъл for.

Безкраен цикъл

С цикъл while също е възможно създаването на безкрайни цикли, тук идеята е условието да е винаги изпълнено:

```
while( 1 )
{
	// Тяло на брезкрайния цикъл
}
```

Цикъл DO-WHILE

Начин на употреба

```
<инициализация>;
do
{
	//Тяло
	<актуализация>;
}
while(<условие>);
```

Цикълът от тип do-while е единственият в програмния език С, който е с пост условие. В този случай тялото на цикъла ще се изпълни поне веднъж, независимо от това дали е изпълнено условието. Причината за това е, че условието се намира след тялото в кода. При do-while също частите за инициализация и актуализация не са предвидени в синтаксиса.

Примерна задача

Да се напише програмен фрагмент, който да отпечата числата от 1 до 10 на екрана.

```
int i;
i=1;
do
{
	printf("%d ",i);
	i = i + 1;
}
while(i<=10);
```

Съкратено записване на някои оператори

Запис	Значение
A++ или ++A ***(не са еднозначни съвсем)***	A=A+1
A--или --A ***(не са еднозначни съвсем)***	A=A-1
A+=5	A=A+5
A-=5	A=A-5
A*=5	A=A*5
A/=5	A=A/5
A%=5	A=A%5
A&=5	A=A&5
A|=5	A=A|5
A^=5	A=A^5
A<<=2	A=A<<2
A>>=2	A=A>>2

Съкратеното записване на някои оператори спомага за по – четлив код, както и работи по – бързо от нормалния запис. Има особености при времето на изпълнение на действието, като трябва да се внимава при комбинации между отделните оператори.

Задачи за упражнение

1. Да се напише програма, която прочита от клавиатурата едно цяло число и изкарва числова пирамида:
   1
   2 2
   3 3 3
   .....
2. Да се напише програма, която прочита от клавиатурата две цели числа и извежда сумата на числата между тези две числа. Въведените числа от клавиатурата не е задължително да са в ред по-малко, по-голямо.
   Вход: 2 7
   Изход: 18 Обяснение 3+4+5+6 = 18
3. Да се напише програма, която прочита от клавиатурата две числа и извежда сумата на четните числа и произведението на нечетните, които се намират между тези две числа.
4. Да се напише програма, при която се въвеждат две числа N и K. След това се въвеждат още N числа и се извежда броят на числата, които са по-големи от K и се делят на 3.
5. Да се напише програма, която при въвеждането на произволен брой числа извежда сборът им. За край на въвеждането се приема въвеждането на 0.
6. Да се напише програма, която прочита от клавиатурата две реални числа и извежда стойностите на фукция f(x) = x*x-4 за всички стойности в дадения интервал. Стъпката на обхождане е 0.01