



Instrucțiunea

While ()

Vs

Do { } while ()

01 Sintaxa generală

Instrucțiunea While

conține o expresie booleană, numită și condiție, care controlează execuția repetată a unei instrucțiuni. Această instrucțiune se execută repetat atât timp cât condiția este adevărată.

Instrucțiunea
WHILE

Sintaxa generală

While <Expresie booleană>
 <Instrucțiune>



Instrucțiunea Do ...While

indică repetarea unei secvențe de instrucțiuni în funcție de valoarea unei expresii booleene. Secvența de instrucțiuni se va executa repetat atât timp cât condiția este true.

Instrucțiunea
Do ... While

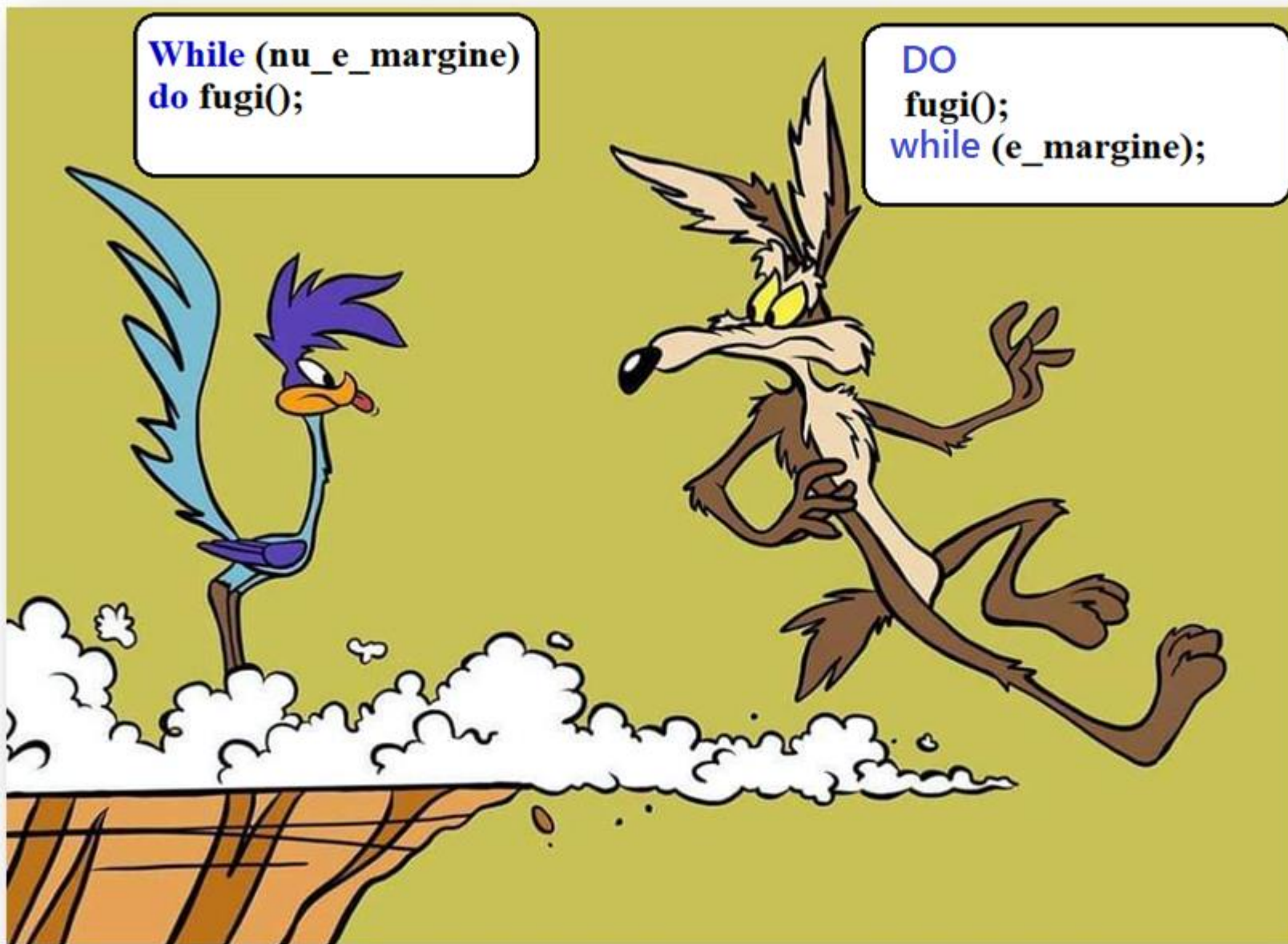
Sintaxa generală

Do
 {<Instrucțiune>}
While <Expresie booleană>



02 Reprezentare vizuală

Instrucțiunile repetitive **While** & **Do...while** sunt utile atunci când numărul de repetiții este nedefinit.



02 Modul de execuție



- 1 Se evaluează condiția logică;
- 2 Dacă valoarea condiției este *adevărat* (true), se execută instrucțiunea aflată între { }, apoi se revine la pasul 1;
- 3 Dacă valoarea condiției este *fals* (false), se continuă cu următoarea instrucțiune aflată după instrucțiunea while.

Sintaxa generală

While <Expresie booleană>
<Instrucțiune>

```
scanf("%d", &n);  
While (n!=0) { scanf("%d", &n); }
```



Dacă valoarea condiției este *fals* (false) de la început atunci instrucțiunea ce se află după { } nu se va executa.

While se numește instrucțiune cu **test inițial** sau condiționată **anterior**.

03 Exemplul #1

#2 Se consideră un număr natural N . Să se scrie un program care numără câte cifre conține acest număr și calculează suma lor.

Input: Valoarea numărului N .

Output: Numărul de cifre din componența numărului dat (k) și suma lor (sum).

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n, count = 0, sum = 0;
```

```
    scanf("%d", &n);
```

```
    while (n != 0) {
```

```
        int rest = n % 10;
```

```
        sum += rest;
```

```
        count++;
```

```
        n /= 10;
```

```
    }
```

```
    printf("%d cifre; sum = %d", count, sum);
```

```
    return 0;
```

```
}
```

32541 | 10
32540 | 3254
restul 1

restul 1

3254 | 10
3250 | 32
rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

rest 4

4^n	rest	sum	count
5	10	3	
2	12	4	
3	15	5	

Consolă

count=5; sum=15

$c\hat{a}t \leftarrow N / 10$

$rest \leftarrow N \% 10$

03 Exemplul #2



#2 Se citesc numere de la tastatură până la apariția lui zero. Să se determine câte dintre numerele citite sunt pare.

Input: Valoarea numărului N .

Output: Numărul de cifre din componența numărului dat (k) și suma lor (sum).

```
#include <stdio.h>
int main(void){
    int nr, count = 0;
    scanf("%d", &nr);
    while (nr != 0) {
        if (nr % 2 == 0) {
            count++;
        }
        scanf("%d", &nr);
    }
    printf("Sunt %d numere pare", count);
    return 0;
}
```

Cerințe: Modificați codul astfel încât să se numere doar numerele impare.

- Calculați suma tuturor numerelor citite de la tastatură.
- Calculați media aritmetică doar a numerelor pare citite de la tastatură.

03 Exemplul #3

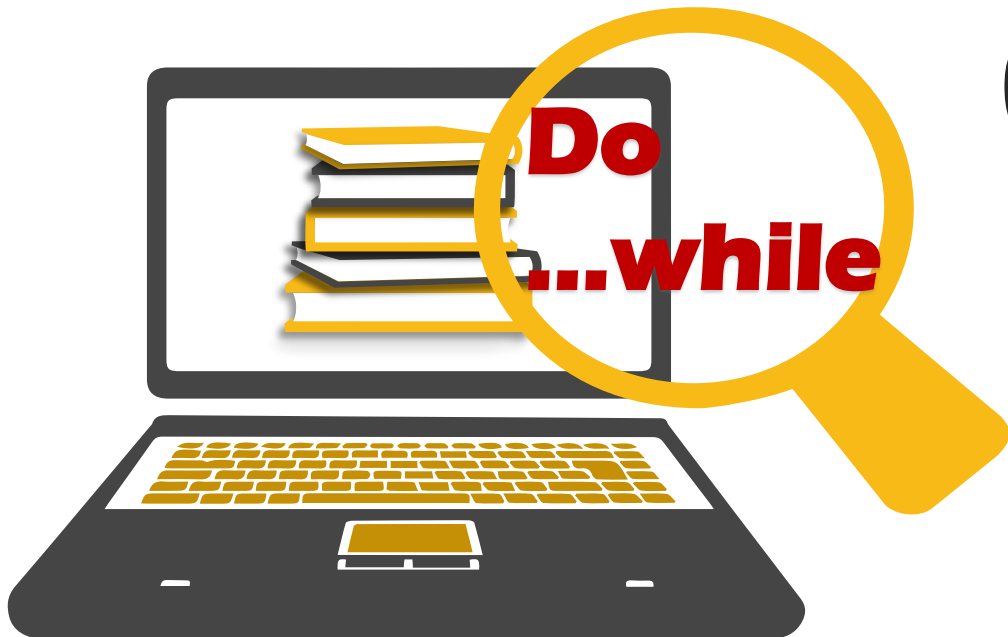


#2 Să se scrie un program care să determine suma cifrelor unui număr natural citit de la tastatură.

```
#include <stdio.h>
int main(void) {
    int nr, sum = 0, rest;
    scanf("%d", &nr);
    do{
        rest = nr % 10;
        sum += rest;
        nr /= 10;
    } while (nr != 0);
    printf("Suma cifrelor este %d", sum);
    return 0;
}
```

Cerințe:

- Modificați codul astfel încât să se numere câte cifre are numărul citit de la tastatură.
- Modificați astfel încât numărul să fie generat random.
- Calculați media aritmetică a cifrelor numărului generat random.



02 Modul de execuție

- 1 Se execută secvența de instrucțiuni;
- 2 Se evaluează condiția logică;
- 3 Dacă valoarea condiției este *true*, se revine la pasul 1;
- 4 Dacă valoarea condiției este *False*, se continuă cu următoarea instrucțiune aflată după instrucțiunea DO, adică după punct și virgulă.

Sintaxa generală

Do

{<Instrucțiune>}

while <Expresie booleană>

Do

```
{ scanf("%d", &n); }
```

```
while (n=0);
```



Do...
while

Indiferent de valoarea condiției, secvența de instrucțiuni dintre Do și While se va executa cel puțin o dată.

se numește instrucțiune cu test final sau condiționată posterior.

04 Extindere

While vs Repeat



Se consideră numărul natural n . Să se scrie un program care calculează suma primilor n termeni:

$$1 + \sqrt{2} + \sqrt{2 \cdot 3} + \dots + \sqrt{2 \cdot 3 \cdot \dots \cdot n} .$$

4

Se consideră trei numere naturale n, k, r . Să se scrie un program care determină primele n numere care împărțite la k , obținem restul r .

2

3

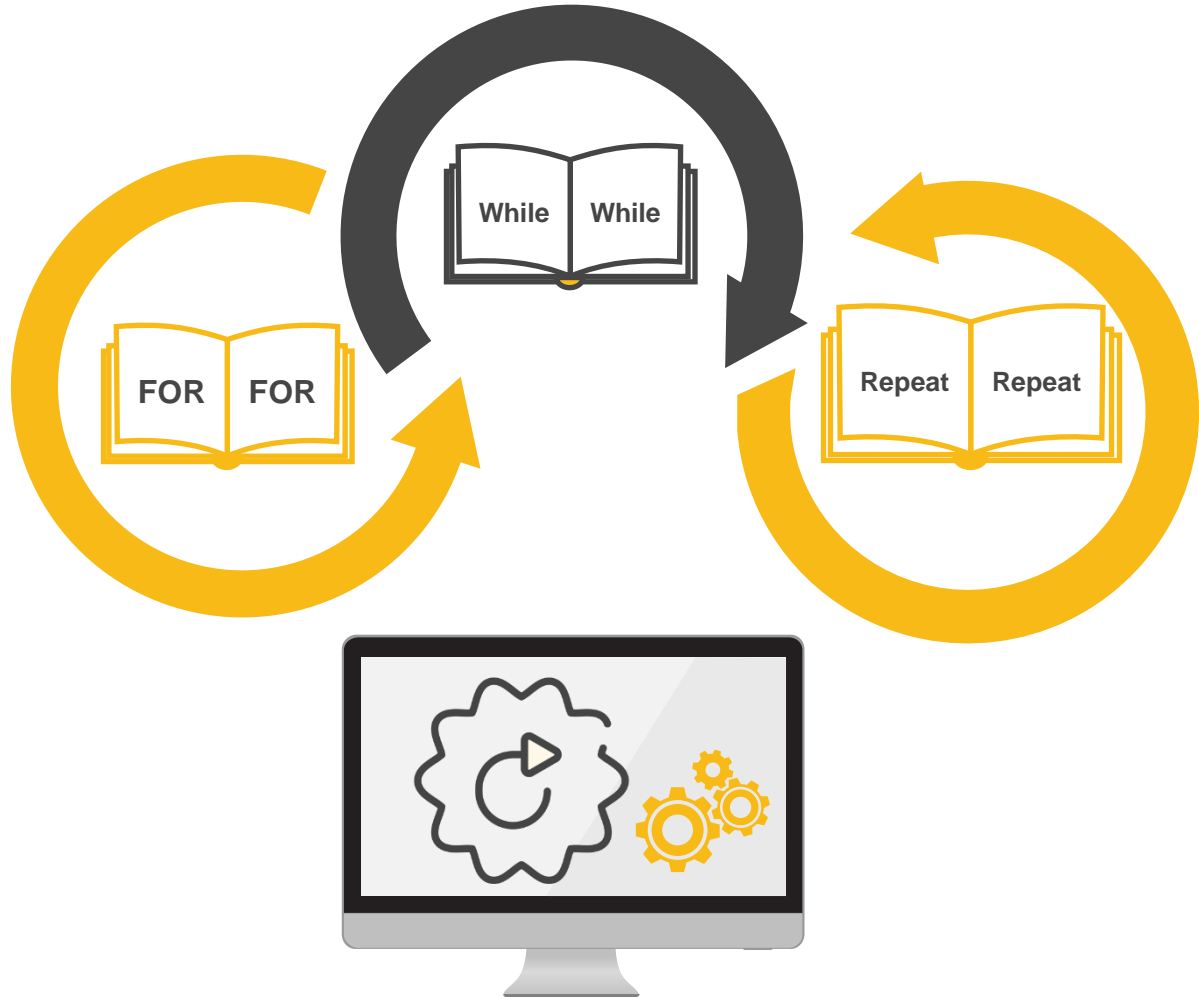
Se consideră numărul natural n . Să se scrie un program care determină primul număr prim ce va urma după numărul n .

1

Se consideră numărul natural n . Să se scrie un program care calculează suma primilor n termeni:
 $1 + \sqrt{2} + \sqrt{3} + \dots + \sqrt{n} .$



Sinteză



FOR este o instrucțiune repetitivă cu un număr cunoscut de pași, adică numărul de repetiții este definit.

While este o instrucțiune repetitivă cu test inițial. Numărul de repetări este nedefinit.

Do ... While este o instrucțiune repetitivă cu test final. Numărul de repetări este nedefinit.



Vă mulțumesc
pentru atenția acordată!