



ALGORITMOS E PROGRAMAÇÃO

Apresentação 06

Roteiro

- Apresentação dos Objetivos
- Levantamento de conhecimentos prévios
- Resolução de exercícios das aulas anteriores
- Estrutura de Repetição While (Enquanto)
- Atividade Prática

ESTRUTURA DE REPETIÇÃO: WHILE

Imagine que você precisa desenvolver um programa C que imprima os números pares de 0 até 4.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int pares;
6
7     printf("Numeros Pares:\n");
8     pares = 0;
9     printf("%d\n", pares);
10    pares = 2;
11    printf("%d\n", pares);
12    pares = 4;
13    printf("%d\n", pares);
14    return 0;
15 }
```

Teste de Mesa

pares
0
2
4

ESTRUTURA DE REPETIÇÃO: WHILE

Imagine que você precisa desenvolver um programa C que imprima os números pares de 0 até 4.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int pares;
6
7     printf("Numeros Pares:\n");
8     pares = 0;
9     printf("%d\n", pares);
10    pares = pares + 2;
11    printf("%d\n", pares);
12    pares = pares + 2;
13    printf("%d\n", pares);
14    return 0;
15 }
```

Teste de Mesa

pares
0
0 + 2 = 2
2 + 2 = 4

Contadores – Passo 2

ESTRUTURA DE REPETIÇÃO: WHILE

Imagine que você precisa desenvolver um programa C que imprima os números pares de 0 até 20.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     int pares;
6
7     printf("Numeros Pares:\n");
8     pares = 0;
9     printf("%d\n", pares);
10    pares = pares + 2;
11    printf("%d\n", pares);
12    pares = pares + 2;
13    printf("%d\n", pares);
14    pares = pares + 2;
15    printf("%d\n", pares);
16    pares = pares + 2;
17    printf("%d\n", pares);
18    pares = pares + 2;
19    printf("%d\n", pares);
20    pares = pares + 2;
21    printf("%d\n", pares);
22    pares = pares + 2;
23    printf("%d\n", pares);
24    pares = pares + 2;
25    printf("%d\n", pares);
26    pares = pares + 2;
27    printf("%d\n", pares);
28    pares = pares + 2;
29    printf("%d\n", pares);
30
31    return 0;
32 }
```

ESTRUTURA DE REPETIÇÃO: WHILE

Imagine que você precisa desenvolver um programa C que imprima os números pares de 0 até 1000.




ESTRUTURA DE REPETIÇÃO: WHILE

Sempre que necessitamos repetir comandos podemos utilizar uma estrutura de repetição.

algoritmo

```
enquanto <condição> faça  
    C1;  
    C2;  
    .  
    .  
    .  
    Cn  
fimenquanto;
```

C



```
while (condicao)  
{  
    C1;  
    C2;  
    ...  
    Cn;  
}
```

ESTRUTURA DE REPETIÇÃO: WHILE

Imagine que você precisa desenvolver um programa C que imprima os números pares de 0 até 1000.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      int pares;
6
7      printf("Numeros Pares:\n");
8      pares = 0;
9      while(pares <= 1000)
10     {
11         printf("%d\n", pares);
12         pares = pares + 2;
13     }
14     return 0;
15 }
```

Enquanto a Condição for VERDADEIRA, a Lista de Comandos é executada.

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 2: Crie um programa C que faça o somatório do peso de 5 pessoas.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      float peso, somatorio;
6      int nropessoas;
7
8      somatorio = 0;
9      nropessoas = 1;
10     printf(":::::Somatorio dos Pesos:::::");
11     while(nropessoas<=5)
12     {
13         printf("\nInforme o peso:");
14         scanf("%f",&peso);
15         somatorio = somatorio + peso;
16         nropessoas = nropessoas + 1;
17     }
18     printf("O Somatorio dos Pesos de 5 pessoas e: %0.2f", somatorio);
19     return 0;
20 }
```

ACUMULADOR

CONTADOR

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 2: Crie um programa C que faça o somatório do peso de 5 pessoas.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      float peso, somatorio;
6      int nropessoas;
7
8      somatorio = 0;
9      nropessoas = 1;
10     printf(":::::Somatorio dos Pesos:::::");
11     while(nropessoas<=5)
12     {
13         printf("\nInforme o peso:");
14         scanf("%f",&peso);
15         somatorio = somatorio + peso;
16         nropessoas = nropessoas + 1;
17     }
18     printf("O Somatorio dos Pesos de 5 pessoas e: %0.2f", somatorio);
19     return 0;
20 }
```

Peso	Somat.	NroPes.

Teste de Mesa

Quantas repetições serão realizadas?

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 2: Crie um programa C que faça o somatório do peso de 5 pessoas.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      float peso, somatorio;
6      int nropessoas;
7
8      somatorio = 0;
9      nropessoas = 1;
10     printf(":::::Somatorio dos Pesos:::::");
11     while(nropessoas<=5)
12     {
13         printf("\nInforme o peso:");
14         scanf("%f",&peso);
15         somatorio = somatorio + peso;
16         nropessoas = nropessoas + 1;
17     }
18     printf("O Somatorio dos Pesos de 5 pessoas e: %0.2f", somatorio);
19     return 0;
20 }
```

Peso	Somat.	NroPes.
	0	1
80	80	2
100	180	3
50	230	4
80	310	5
90	400	6

Teste de Mesa

Quantas repetições serão realizadas?

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 3: Crie um programa C que leia valores até que seja digitado um número múltiplo de 5. Se não for múltiplo de 5 diga “Não é múltiplo de 5, informe novamente.”

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      int num;
7
8      printf("Informe um numero multiplo de 5: ");
9      scanf("%d",&num);
10     while(num % 5 !=0)
11     {
12         printf("\nNao e multiplo de 5, informe novamente!!!");
13         printf("\nInforme um numero multiplo de 5: ");
14         scanf("%d",&num);
15     }
16     printf("O numero %d e multiplo de 5!!!",num);
17 }
```

Quantas repetições serão realizadas?

num	Impressão

Teste de Mesa

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 3: Crie um programa C que leia valores até que seja digitado um número múltiplo de 5. Se não for múltiplo de 5 diga “Não é múltiplo de 5, informe novamente.”

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      int num;
7
8      printf("Informe um numero multiplo de 5: ");
9      scanf("%d",&num);
10     while(num % 5 !=0)
11     {
12         printf("\nNao e multiplo de 5, informe novamente!!!");
13         printf("\nInforme um numero multiplo de 5: ");
14         scanf("%d",&num);
15     }
16     printf("O numero %d e multiplo de 5!!!",num);
17 }
```

Quantas repetições serão realizadas?

num	Impressão
	Informe um numero multiplo de 5:
7	Nao e multiplo de 5, informe novamente!!!
	Informe um numero multiplo de 5:
13	Nao e multiplo de 5, informe novamente!!!
	Informe um numero multiplo de 5:
27	Nao e multiplo de 5, informe novamente!!!
	Informe um numero multiplo de 5:
41	Nao e multiplo de 5, informe novamente!!!
	Informe um numero multiplo de 5:
90	O numero 90 e multiplo de 5!!!

Teste de Mesa

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 4: Crie um programa C que pergunte quantos números aleatórios devem ser gerados e qual seu limite máximo (maior valor)."

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      int quantidade, num, maior;
7
8      printf("Informe quantos numeros aleatorios voce deseja:");
9      scanf("%d",&quantidade);
10     printf("Informe limite maximo(maior valor):");
11     scanf("%d",&maior);
12     srand(time(NULL)); // srand -> cada execução traz números diferentes
13     while(quantidade > 0)
14     {
15         num = rand() % maior; //rand() -> sorteia número
16         printf(" %d - ", num);
17         quantidade--;
18     }
19 }
```

quantidade	maior	número

Quantas repetições serão realizadas?

Teste de Mesa

ESTRUTURA DE REPETIÇÃO: WHILE

Exemplo 4: Crie um programa C que pergunte quantos números aleatórios devem ser gerados e qual seu limite máximo (maior valor)."

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      int quantidade, num, maior;
7
8      printf("Informe quantos numeros aleatorios voce deseja:");
9      scanf("%d",&quantidade);
10     printf("Informe limite maximo(maior valor):");
11     scanf("%d",&maior);
12     srand(time(NULL)); // srand -> cada execução traz números diferentes
13     while(quantidade > 0)
14     {
15         num = rand() % maior; //rand() -> sorteia número
16         printf(" %d - ", num);
17         quantidade--;
18     }
19 }
```

quantidade	maior	número
7	100	67
6		98
5		21
4		56
3		11
2		87
1		44
0		

Quantas repetições serão realizadas?

Teste de Mesa

DICA 1:

Na linguagem C o comando MOD (resto da divisão de inteiros) é **%**.

Ex: `r = 5 % 2`, ou seja o resto da divisão de 5 por 2 é 1

DICA 2:

Na linguagem C o símbolo de “diferente de” é **!=**

Ex: `if(nota != 10)`, ou seja, se nota diferente de 10.

DICA 3:

Na linguagem C a comparação se um número é igual ao outro é feita com 2 sinais de igual **==**

Ex: `if(nota == 10)`, ou seja, se nota igual a 10.

DICA 4:

Na linguagem C, quando você precisa fazer cálculos entre variáveis inteiras e reais (float) é necessário converter as variáveis reais para float colocando à esquerda da variável inteira o comando **(float)**

Ex:

```
float mediaAltura, somaAlturas;  
int numeroAlunos;
```

```
mediaAltura = somaAlturas / (float)numeroAlunos;
```

ATIVIDADES PRÁTICAS

Crie os seguintes códigos em linguagem C e depois faça o teste de mesa:

1. *Crie um programa em C que leia um número e imprima a sua tabuada.*
2. *Crie um programa em C que leia dois números pares e positivos e imprima todos os pares entre eles.*
3. *Crie um programa em C que leia dois números pares e positivos e imprima a soma de todos os pares entre eles.*
4. *Crie um programa em C que leia dois números pares e positivos e imprima a média aritmética de todos os pares entre eles.*
5. *Crie um programa em C que leia um número inteiro e imprima o fatorial desse número.
Ex.: fatorial de 5 = $5*4*3*2*1$, ou seja 120*

ATIVIDADES PRÁTICAS

Crie os seguintes códigos em linguagem C e depois faça o teste de mesa:

6. *Crie um programa em C que leia a altura e o sexo de 20 alunos (1- Masc. 2-Fem.) e informe :*
 - a) *Número de alunos*
 - b) *Número de alunas*
 - c) *Média de altura dos alunos*
 - d) *Média de altura das alunas*

7. *Crie um programa em C que leia a idade dos alunos de uma turma e depois calcule a idade do mais velho e a idade do mais novo.*

Obs.: Para finalizar a leitura das idades, digite um valor negativo para idade.



ecossistema
ănima