Estruturas de Controle de Fluxo

3. Laços de repetição

Exercício 6:

Faça um programa que receba números naturais fornecidos pelo usuário, quando o usuário quiser parar a execução do programa, o mesmo fornecerá um número negativo. O programa deve retornar ao final a média dos números naturais fornecidos pelo usuário (fazer dois programas utilizando em cada um uma das estruturas de repetição vistas).

UNIVASF

Estruturas de Controle de Fluxo

3. Laços de repetição

Exercício 6:

Faça um programa que receba números naturais fornecidos pelo usuário, quando o usuário quiser parar a execução do programa, o mesmo fornecerá um numero negativo. O programa deve retornar ao final a média dos números naturais fornecidos pelo usuário (fazer dois programas utilizando em cada um uma das estruturas de repetição vistas).

UNIVASF

```c
#include <stdio.h>
int main() {
   int num,contador=0, acumulador=0;
   do {
      printf ("Entre com um numero natural (entre com um numero negativo para sair):");
      scanf ("%d",&num);
      if (num>=0) {
         acumulador += num;
         contador++;
      }
   } while (num>=0);
   printf ("A media dos %d numeros naturais pelo usuario eh %f", contador,
   acumulador/(float)contador);
}
```

Se o primeiro valor fornecido for negativo?

```c
#include <stdio.h>
int main() {
   int num,contador=0, acumulador=0;
   do {
      printf ("Entre com um numero natural (entre com um numero negativo para sair):");
      scanf ("%d",&num);
      if (num>=0) {
         acumulador += num;
         contador++;
      }
   } while (num>=0);
```

```c
#include <stdio.h>
int main() {
   int num,contador=-1,acumulador;
   num=acumulador=0;
   while (num>=0) {
      contador++;
      printf ("Entre com um numero natural (entre comum numero negativo para sair):");
      scanf ("%d",&num);
      if (num>=0)
         acumulador += num;
   }
   if (contador)
      printf ("A media dos %d numeros naturais pelo usuario eh %f", contador,
      acumulador/(float)contador);
   else
      printf ("Nao foi fornecido nenhum numero natural");
}
```

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exercício 7:

Faça um programa que receba um número natural fornecido pelo usuário e retorne seus divisores em ordem decrescente na saída padrão. (fazer dois programas distintos utilizando em cada um uma das estruturas de repetição vistas).

Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exercício 7:

Faça um programa que receba um número natural fornecido pelo usuário e retorne seus divisores em ordem decrescente na saída padrão. (fazer dois programas distintos utilizando em cada um uma das estruturas de repetição vistas).

```c
#include <stdio.h>
int main()
{
    int num, div;
    printf ("Digite um numero natural:");
    scanf ("%d", &num);
    div = num;
    do
    {
        if (num%div==0)
            printf ("%d eh divisor de %d\n", div, num);
        div--;
    }while (div);
}
```

Se o usuário fornecer um valor negativo?

```c
#include <stdio.h>
int main()
{
    int num, div;

        printf ("Digite um numero natural:");
        scanf ("%d", &num);

    div = num;   div = num?num:1;
    do
    {
        if (num%div==0)
            printf ("%d eh divisor de %d\n", div, num);
        div--;
    }while (div);
}
```

Se num for igual a zero?

```c
#include <stdio.h>
int main() {
    int num=-1, div;
    while (num<0) {
        printf ("Digite um numero natural:");
        scanf ("%d", &num);
    }
    div = num?num:1;
    while (div)
    {
        if (num%div==0)
            printf ("%d eh divisor de %d\n", div, num);
        div--;
    }
}
```

```c
#include <stdio.h>
int main() {
    int num, div;
    do {
        printf ("Digite um numero natural:");
        scanf ("%d", &num);
    } while (num<0);
    div = num?num:1;
    while (div)
    {
        if (num%div==0)
            printf ("%d eh divisor de %d\n", div, num);
        div--;
    }
}
```

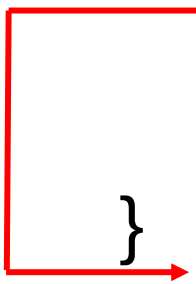
Estruturas de Controle de Fluxo
Comandos break e continue

Estruturas de Controle de Fluxo

4. Comando *break*

Exemplo:

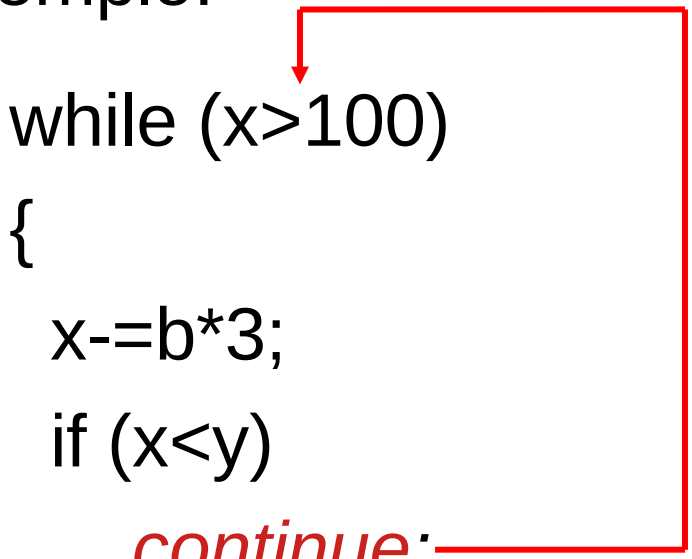
```
while (x>100)
{
        x-=b*3;
        if (x<100)
            break;
        x-=y*3;
}
```



Estruturas de Controle de Fluxo

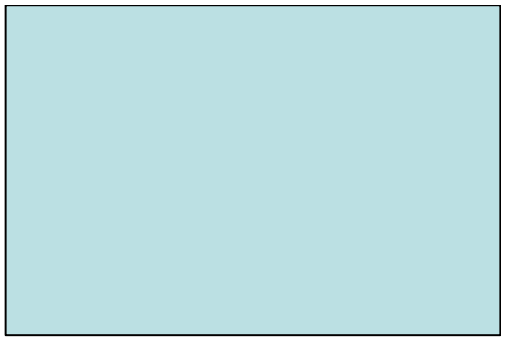
5. Comando *continue*

Exemplo:



```
while (x>100)
{
  x-=b*3;
  if (x<y)
     continue;
  x-=y*3;
}
```

Estruturas de Controle de Fluxo



Visando fomentar uma maior participação de discentes em competições acadêmicas, mais especificamente na Maratona de Programação e, também, buscando viabilizar uma possível correção parcialmente automatizada de uma avaliação da disciplina, trabalharemos com uma forma alternativa para entrada de dados para programas.

Como vimos até o momento os programas ao se utilizarem da função de entrada formatada *scanf*() efetuam leitura de dados através do teclado.

Também vimos que tudo que é digitado é armazenado em um arquivo denominado *stdin* e posteriormente lido pelas chamadas à função *scanf*().

UNIVASF

Estruturas de Controle de Fluxo

Para uma melhor compreensão, vamos analisar o programa a seguir:

```c
#include <stdio.h>
int main ()
{
   int valorUm, valorDois;
   printf("Digite um valor inteiro: ");
   scanf("%d", &valorUm);
   printf("Digite outro valor inteiro: ");
   scanf("%d", &valorDois);
   printf("%d+%d=%d\n", valorUm, valorDois,
   valorUm+valorDois);
}
```

Estruturas de Controle de Fluxo

Agora abra um editor de texto e digite no mesmo o conteúdo abaixo:

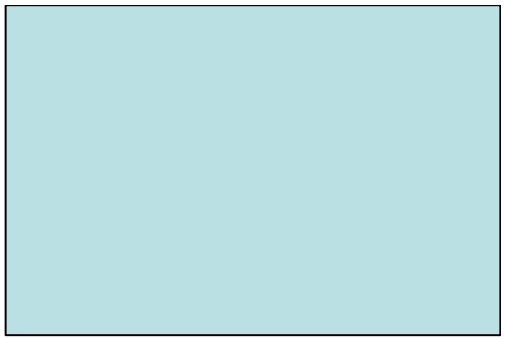
6 13

Após, salve o texto em um arquivo com o nome entrada.txt na pasta em que você salvou o arquivo fonte.

No prompt de comando (no Windows) ou no terminal (no Linux), na pasta em que você compilou o arquivo fonte digite:

./nomeDoArquivoBinario < entrada.txt

Estruturas de Controle de Fluxo



Será obtida a seguinte saída no monitor:

Digite um valor inteiro: Digite outro valor inteiro: 6+13=19

Isto demonstra que podemos através do símbolo < informar um arquivo que será considerado como *stdin.*

Vamos agora editar o código fonte retirando os trechos de solicitação de entradas por parte do usuário, sublinhados em vermelho no próximo slide.

UNIVASF

Estruturas de Controle de Fluxo

```c
#include <stdio.h>
int main ()
{
   int valorUm, valorDois;
   printf("Digite um valor inteiro: ");
   scanf("%d", &valorUm);
   printf("Digite outro valor inteiro: ");
   scanf("%d", &valorDois);
   printf("%d+%d=%d\n", valorUm, valorDois,
   valorUm+valorDois);
}
```

Após uma nova compilação e execução com:

./nomeDoArquivoBinario < entrada.txt

Temos a seguinte saída:

6+13=19

Estruturas de Controle de Fluxo

Com base no que foi visto, resolva o exercício a seguir:

Construa um programa que receba uma sequência de números inteiros, positivos de três dígitos, e gere como saída, para cada valor pertencente à sequência, outro número formado pelos dígitos invertidos do número lido. O final da sequencia é identificado pelo valor zero.

Exemplo de entrada:	Exemplo de saída:
123	321
901	109
530	035
0	

Dica: Observe os resultados das funções Quociente e Resto de um número por 10.