

# OPERADORES

LICENCIATURA EM ENGENHARIA  
INFORMÁTICA

# OPERAÇÃO COM INTEIROS

- Operações com números inteiros devolvem resultados com valor inteiro.

Operação	Descrição	Exemplo	Resultado
+	Soma	$21 + 4$	25
-	Subtração	$21 - 4$	17
*	Multiplicação	$21 * 4$	84
/	Divisão	$21 / 4$	5
%	Resto da divisão	$21 \% 4$	1

# OPERAÇÃO COM INTEIROS

- Declare a variável *numero* do tipo *int* que inicializada com o valor 2016.

```
#include<stdio.h>
main(){
    int numero = 2016;
    printf("O valor de numero = %d e o valor seguinte = %d\n",
           numero, numero + 1);
    getchar();
    return(0);
}
```

O valor de numero = <íntero> e o valor seguinte = <íntero>

O formato de escrita de um íntero na função *printf* é %d

# OPERAÇÃO COM INTEIROS

- Declare a variável *numero* do tipo *int* que leia um valor inteiro qualquer

```
#include<stdio.h>
main(){
    int numero;
    printf("introduza um número: ");
    scanf("%d", &numero);
    printf("o número introduzido foi %d\n", numero);
    getchar(); getchar();
    return(0);
}
```

Para ler uma variável qualquer do tipo *int*, *char*, *float* ou *double* utilizando a função *scanf*, deve ser precedida do símbolo & (e comercial)

# OPERAÇÃO COM INTEIROS

- Interprete o programa seguinte

```
#include<stdio.h>
int main(void){
    int n1, n2;
    printf("introduza dois números: ");
    scanf("%d%d", &n1,&n2);
    printf("O resultado de %d + %d = %d\n",n1,n2,n1+n2);
    getchar();getchar();
    return(0);
}
```

Para ler uma variável qualquer do tipo int, char, float ou double utilizando a função *scanf*, deve ser precedida do símbolo & (e comercial)

# OPERAÇÃO COM INTEIROS

- Interprete o programa seguinte

```
#include <stdio.h>
main ()
{
    char c,c1,c2,c3,c4,c5;
    printf ("Escreva IMETRO: "); /* Entrada de Dados */
    scanf("%c%c%c%c%c", &c,&c1,&c2,&c3,&c4,&c5);
    printf (" o conjunto de caracteres sera '%c\n%c\n%c\n%c\n%c\n%c\n",
c,c1,c2,c3,c4,c5);

    getchar();getchar();
    return(0);
}
```

Para ler vários caracteres de forma sequencial usando a função  
*scanf()*

# OPERADOR SIZEOF

- Retorna a dimensão de um inteiro (outro tipo de dados ou variável)
- *O programa abaixo retorna o numero de bytes que ocupa um tipo de dados*

```
#include<stdio.h>
int main(void){
    printf("O tamanho em bytes de um inteiro = %d\n",sizeof(int));
    printf("O tamanho em bytes de um inteiro = %d\n",sizeof(float));
    printf("O tamanho em bytes de um inteiro = %d\n",sizeof(double));
    printf("O tamanho em bytes de um inteiro = %d\n",sizeof(char));
    printf("O tamanho em bytes de um inteiro = %d\n",sizeof(long));
    getchar();
    return(0);
}
```

**sizeof <expressão> ou sizeof (<tipo>)**

# SHORT E LONG

- Escrever um programa que solicite ao utilizador a idade, o montante a depositar e o nº de conta em que se quer realizar o depósito, declarando as variáveis como *short*, *int* e *long*.

```
#include<stdio.h>
int main(void){
    short idade;
    int montante;
    long n_conta;
    printf("Qual a idade: "); scanf("%hd", &idade);
    printf("Qual o montante a depositar: "); scanf("%d", &montante);
    printf("Qual o nº de conta: "); scanf("%ld", &n_conta);
    printf("Uma pessoa de %hd anos depositou %dkz00 na conta
%ld\n: ",idade, montante, n_conta);
    getchar();getchar();
    return(0);
}
```

Qual a idade: 42

Qual o montante a depositar: 36000

Qual o numero de conta: 123456789

Uma pessoa de 42 anos depositou 36000\$00 na conta 123456789

# PREFIXOS SIGNED E UNSIGNED

- O prefixo *signed* antes de um inteiro, não é necessário pois por defeito todos os inteiros quando são criados são sinalizados (*signed*)

```
#include<stdio.h>
int main(void){
    unsigned int idade; /*ou unsigned idade*/
    getchar();getchar();
    return(0);
}
```

No formato de leitura e escrita de variáveis inteiras sem sinal (**unsigned int**), utilizando as funções scanf e printf é %u em vez de %d

SIGNED E UNSIGNED

# OPERADORES FLOAT E DOUBLE

- Usadas para valores numéricos com partes fracionárias (reais ou de virgula flutuante)

```
#include<stdio.h>
int main(void){
    float raio, perimetro;
    double Pi = 3.1415927, area;
    printf("Introduza o raio da circunferencia: ");
    scanf("%f",&raio);
    area = Pi * raio * raio;
    perimetro = 2 * Pi * raio;
    printf("Area      =      %f\nPerimetro      =%f\n",      area,      perimetro);
    getchar();getchar();
    return(0);
}
```

Programa que calcule a área da circunferência

# OPERADORES COM CARACTERES

- O tipo *char* permite armazenar uma variável deste tipo.
- *Um único caracter*
- Armazenada num único byte

```
#include<stdio.h>
int main(void){
    printf("%cmetro %cngola", 'I','A','\n');
    getchar();
    return(0);
}
```

A sua representação é feita utilizando plicas ('a')  
O seu formato de leitura é %c

CHAR

# OPERADORES COM CARACTERES

- Escreva um programa que faça a leitura de um caracter do teclado e retorne na consola.

```
#include<stdio.h>
int main(void){
    char a;
    printf("Introduza um caracter: ");
    scanf("%c", &a);
    printf("O caracter introduzido foi '%c'\n", a);
    getchar();
    return(0);
}
```

CHAR

A sua representação é feita utilizando plicas ('a')  
O seu formato de leitura é %c

# CASTING - CONVERSÃO

- Serve para modificar o valor de um determinado tipo de dado para outro.

```
#include<stdio.h>
int main(void){
    float a;
    printf("Introduza um numero inteiro: ");
    scanf("%f",&a);
    printf("O numero introduzido foi %f, cujo inteiro = %d\n", a,(int)a);
    system("pause");
    return(0);
}
```

CASTING

O tipo pretendido será colocado entre parêntesis antes do valor

# EXERCÍCIOS

- Escreva um programa que solicite ao utilizador dois valores inteiros e apresente o resultado da realização das operações aritméticas tradicionais.

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    int a,b; /* Declaracao de Variaveis */
    printf ("digite dois inteiros: "); /* Entrada de Dados */
    scanf ("%d%d",&a,&b);
    printf("%d + %d = %d\n",a,b, a+b);
    printf("%d - %d = %d\n",a,b, a-b);
    printf("%d * %d = %d\n",a,b, a*b);
    printf("%d / %d = %d\n",a,b, a/b);
    printf("%d %% %d = %d\n",a,b, a%b);
    getchar();
    getchar();
    return(0);
}
```

# EXERCÍCIOS

# EXERCÍCIOS

- Escreva um programa que solicite um determinado numero de segundos e em seguida indique quantas horas, minutos e segundos esse valor representa.
  - Considere horas( $\text{segundos} / 3600$ )
  - Considere minutos ( $\text{Segundos} / 60$ )
  - Considere segundos ( $\text{segundos \% } 60$ )

# EXERCÍCIOS

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    long int n_segundos; /* Declaracao de Variaveis */
    printf ("introduza o numero de segundos: "); /* Entrada de Dados */
    scanf ("%ld",&n_segundos);
    printf("Horas: %d\n",(int)n_segundos/3600);
    printf("Minutos: %d\n",(int)n_segundos%3600/60);
    printf("Segundos: %d\n",(int)n_segundos%60);
    getchar();
    getchar();
    return(0);
}
```

# EXERCÍCIOS

# EXERCÍCIOS

- Escreva um programa que solicite um determinado numero real e mostre qual a sua parte fracionaria e a sua parte inteira.

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    float x; /* Declaracao de Variaveis */
    printf ("introduza o numero real: "); /* Entrada de Dados */
    scanf ("%f",&x);
    printf("Parte inteira: %d\n", (int)x);
    printf("Parte Fraccionaria: %f\n", x-(int)x);
    getchar();
    getchar();
    return(0);
}
```

# EXERCÍCIOS

# EXERCÍCIOS

- Escreva um programa que solicite ao utilizador uma determinada data e a mostre em seguida no formato dd/mm/aa.
- Escreva um programa que solicite ao utilizador uma determinada data no formato aaaa-mm-dd e a mostre em seguida no formato dd/mm/aa.
- Escreva um programa que solicite um inteiro (entre 0 e 255) ao utilizador e mostre o inteiro seguinte e o carácter correspondente.

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    int num; /* Declaracao de Variaveis */
    printf ("digite um inteiro: "); /* Entrada de Dados */
    scanf ("%d",&num);
    printf ("Foi introduzido um inteiro %d cujo caracter = '%c' \n", num,
(char)num);
    printf ("o caracter seguinte = '%c' tem o ASCII nº %d \n",
(char)num+1, num+1);
    getchar();
    getchar();
    return(0);
}
```

Digite um inteiro: 67

Foi introduzido o 67 cujo caracter = 'C'

O caracter seguinte = 'D' tem o ASCII nº 68

# EXERCÍCIOS

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    int x, y, x; /* Declaracao de Variaveis */
    float int = 5.23;
    printf ("%f:", int); /* saída de Dados */
    return(0);
}
```

```
#include <stdio.h>
int main ()
{
    int x=y=z=1; /* Declaracao de Variaveis */
    printf ("%d%d%d \n", x,y,z); /* saída de Dados */
    return(0);
}
```

Identifique os erros de compilação que seriam detectados nos programas acima

# EXERCÍCIOS

```
#include <stdio.h>
int main ()
{
    int num; /* Declaracao de Variaveis */
    printf ("digite um inteiro: "); /* Entrada de Dados */
    scanf ("%d \n",&num);
    printf ("Foi introduzido um inteiro %d \n", num);
    return(0);
}
```

```
#include <stdio.h>
int main ()
{
    int num; /* Declaracao de Variaveis */
    printf ("digite um inteiro: "); /* Entrada de Dados */
    scanf ("%f",&num);
    printf ("Foi introduzido um inteiro %f \n", num);
    return(0);
}
```

Identifique os erros de compilação que seriam detectados nos programas acima

# CONTROLO DE FLUXO

LICENCIATURA EM ENGENHARIA  
INFORMÁTICA