Explorando os tipos básicos de dados

Professores(as):
Virgínia Fernandes Mota
João Eduardo Montandon de Araujo Filho
Leandro Maia Silva

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA



- Esse tipo de dado é utilizado para armazenar valores que pertencem ao conjunto dos números inteiros;
- Esse tipo de dado geralmente é armazenado utilizando complemento de 2;
- Geralmente utiliza 32 bits ou 4 bytes, porém esse número pode variar dependendo da arquitetura do computador. Para averiguar a quantidade de bytes que um inteiro gasta, utilize a função "sizeof":

```
1 ... printf("O Tamanho em bytes de um int = %d\n", sizeof(int));
...
```

Um inteiro com 32 bits pode armazenar valores entre
 -2147483648 e 2147483647. Se tentarmos armazenar um
 valor maior que 2147483647 ocorre um overflow (o valor fica
 aparentemente sem sentido). E caso tente assinalar à variável
 um valor menor que -2147483648 ocorre um underflow (o
 valor também fica aparentemente sem sentido).

Os inteiros podem realizar as seguintes operações:

- + adicão;
- subtração;
- * multiplicação;
- / divisão inteira (21/4 = 5);
- % resto da divisão inteira (21%4 = 1).

Modificadores opcionais:

- signed (inteiro com sinal : -2^{n-1} a $2^{n-1}-1$);
- unsigned (inteiro sem sinal 0 a $2^n 1$);
- short (inteiro pequeno 2bytes);
- long (inteiro grande 4bytes).

Tipo - Flutuante

- Geralmente utilizam 32 ou 64 bits;
- Eles são armazenados em notação binária científica:
 - $(-1)^{sinal} \times mantissa \times 2^{expoente}$
- float (4 bytes);
- double (8 bytes);
- long double (em geral, 12 ou 16 bytes).
- A norma que define como os número de ponto flutuante são representados é IEEE754.

Tipo - Flutuante

Os números de ponto flutuante podem realizar as seguintes operações:

- + adição;
- subtração;
- * multiplicação;
- / divisão real (21.0/4.0 = 5.25);

Trabalhando com números - Funções Matemáticas

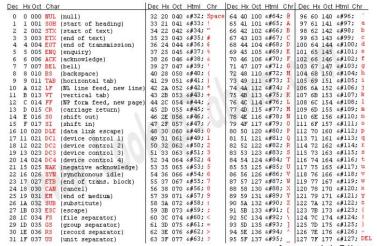
- Com a biblioteca math.h podemos encontrar facilmente funções para calcular potências, raiz quadrada, funções trigonométricas para cálculos que envolvem seno, cosseno e tangente, além de constantes para números irracionais como, por exemplo PI;
- Trigonométricas:
 - sin(): retorna o valor do seno;
 - cos(): retorna o valor do cosseno;
 - tan(): retorna o valor da tangente;

Trabalhando com números - Funções Matemáticas

- Logarítmicas:
 - log(): retorna o valor do logaritmo na base 2;
 - lo10(): retorna o valor do logaritmo na base 10;
- Potências:
 - pow(): retorna o valor da base elevada ao expoente (2¹⁰ = pow(2,10));
 - sqrt(): retorna o valor da raiz quadrada de um número;

Tipo caractere

- Cada variável possui 1 byte;
- Cada combinação representa um caractere.



Source: www.LookupTables.com

Trabalhando com caracteres

- Dentre as funções encontradas na biblioteca ctype.h, há
 aquelas que modificam o estado da letra (maiúsculas e
 minúsculas) e até mesmo funções que servem para descobrir se
 o que foi digitado é um ponto, vírgula, número, espaço, etc;
- Algumas funções:
 - toupper(): esta função recebe um argumento que deve ser um caractere e retorna o caractere correspondente em formato maiúsculo, se o caractere já for maiúsculo, a função não o modifica;
 - tolower(): esta função recebe um argumento que deve ser um caractere e retorna o caractere correspondente em formato minúsculo, se o caractere já for minúsculo, a função não o modifica

Trabalhando com caracteres

- isalnum(): verifica se o caractere ou inteiro passado como parâmetro é alfanumérico. Isso inclui todos os números e as letras do alfabeto, tanto maiúsculas quanto minúsculas.
- isalpha(): verifica se o caractere ou inteiro passado como parâmetro é alfabético. Isso inclui todas as letras do alfabeto, tanto maiúsculas quanto minúsculas.
- isdigit(): verifica se o caractere ou inteiro passado como parâmetro é um digito. Isso inclui todos os números.
- ispunct(): verifica se o caractere ou inteiro passado como parâmetro é uma pontuação. Isso inclui qualquer tipo de pontuação. Porém, não é capaz de verificar se uma letra é acentuada.
- isspace(): verifica se o caractere ou inteiro passado como parâmetro é um espaço em branco.

Trabalhando com caracteres

- islower(): verifica se o caractere ou inteiro passado como parâmetro é uma letra minúscula
- isupper(): verifica se o caractere ou inteiro passado como parâmetro é uma letra maiúscula
- iscntrl(): verifica se o caractere ou inteiro passado como parâmetro é um caractere de comando. Isso inclui CTRL, ALT, ENTER, BACKSPACE, etc.
- isxdigit(): verifica se o caractere ou inteiro passado como parâmetro é compatível com um número hexadecimal. Isso inclui todos os número (0 - 9) e qualquer letra entre A e F (não importa se minúsculo ou maiúsculo).