

MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

Linguagem de Programação C - retomando...

Tipos padrão em C

Tipos estruturados em C

A linguagem C

```
#include <stdio.h>

... /* bibliotecas e definições globais no
arquivo */

/* protótipos das funções locais */

int main () {
    /* declarações */

    /* comandos */

    return (0);
}

/* funções locais */
```

Alguns exemplos...

- ▶ Dada uma sequência de números inteiros terminada por zero, imprima a soma dos números pares da sequência.
- ▶ Dado um inteiro $n > 0$, decide se existem primos p e q tais que $n = p + q$.
- ▶ Dado $n > 0$ imprima os coeficientes de $(a + b)^n$. Por exemplo, se $n = 4$ deve imprimir 1, 4, 6, 4, 1.

Tipos de dados em C

C não é uma linguagem com tipos fortes. Vamos em seguida mostrar os tipos padrão da linguagem.

Variáveis inteiras

O tipo padrão para variáveis inteiras é o `int`. Armazena um inteiro com sinal. Usualmente uma variável usa 4 bytes, assim, pode armazenar valores no intervalo $[-2^{31}, 2^{31} - 1]$.

- ▶ `unsigned`: o número será positivo.
- ▶ `long`: duplica a representação interna (usualmente 8 bytes).

Exemplo: Escreva uma função que recebe um inteiro $n \geq 0$ e devolve $n!$.

Tipos padrão em C

Variáveis “reais”

O tipo padrão para estas variáveis é `double`. Armazena uma aproximação de um número real usando ponto flutuante.

Exemplo: $1234.5678 = 1.2345678 * 10^3$. As duas partes (mantissa e expoente) são armazenadas.

Cuidado: $x = 1 / n;$

Erros de aproximação são um problema quando tratamos com variáveis deste tipo.

Exemplo: Dado um inteiro $n > 0$, calcular

$$\sum_{i=1}^n 1/i - \sum_{i=n}^1 1/i.$$

Leitura e impressão de variáveis

Para ler valores da entrada padrão usamos a função `scanf` da biblioteca `stdio.h`. Para imprimir na saída padrão, a função usada é `printf`.

A função `scanf` recebe como parâmetro uma string com o formato que deve ser usado para a leitura e os endereços das variáveis em que os valores lidos devem ser guardados.

Exemplo:

```
int v1;
unsigned int v2;
long int v3;
double v4;

scanf ("%d %u %ld %lf", &v1, &v2, &v3, &v4);
```

Há muito mais sobre o formato de leitura e impressão e as funções `scanf` e `printf`. Veja este vídeo sobre o assunto.

Tipos padrão em C

Variáveis com caracteres

Uma variável do tipo `char` armazena um inteiro, correspondente ao valor na tabela ASCII de um caractere.

Exemplo:

```
char car1, car2, car3;  
car1 = 'A';  
car2 = '5';  
scanf ("%c", &car3);  
  
for (car1 = 'A'; car1 <= 'Z'; car1++)  
    printf ("%c-%c\n", car1, car1 - 'A' + 'a');
```

Vetores e matrizes estáticos

Podemos criar variáveis com vários elementos do mesmo tipo, chamado vetores e matrizes. O tamanho deve ser constante, conhecido em tempo de compilação:

Exemplo:

```
int vetint[100];  
double A[10][10];  
long mat[1000][1000];
```

Note que as dimensões dos vetores e matrizes deve ser constante. Usualmente, fazemos algo como:

```
#define MAX 1000  
  
char nome[MAX]
```

Strings

Uma `string` em C é um vetor de caracteres que termina com o caractere `'\0'`. Ao contrário dos outros vetores, podemos ler e imprimir `strings` sem fazer isso para cada caractere.

```
char frase[MAX];  
scanf ("%[\n]", frase);  
printf ("%s", frase);
```

A biblioteca `string.h` tem várias funções que manipulam strings.

Para saber mais...

- ▶ Do livro texto, apêndices B, C e G;
- ▶ Do site do Prof. Feofiloff, a parte “Recursos da linguagem C”;
- ▶ Vídeos sobre `scanf` no canal do Prof. Bruno Ribas.