Introdução à Programação

Prof. Rafael Alves Bonfim de Queiroz
rafael.queiroz@ufop.edu.br

Conteúdo

Linguagens de Programação

- **C**: programação estruturada
- **C++**: primeira linguagem orientada a objetos de sucesso comercial
- **Python**: linguagem de programação interpretada

Programação em Linguagem C

- **Passagem de Parâmetros**: valor, ou seja, cópias de todos os argumentos são feitas em chamadas de função
  - Passar um ponteiro para qualquer argumento que você pretenda modificar dentro do corpo da função
  - O ponteiro para x é denotado por $\&x$, enquanto o valor apontado por p é denotado por $*p$
- **Tipos de dados**:
  - *Int* e *float* de maior precisão são indicados como *long* e *double*, respectivamente
  - Todas as funções retornam um valor do tipo *int* se não for especificado de outra forma

Programação em Linguagem C

1. **Arrays**: os índices do array C/C++/Python sempre variam de 0 a $n-1$, onde n é o número de elementos no array
2. **Operadores**: aritméticos $(+, -, /, *, \%)$e lógicos (&&, ||, ==, =)

Dicas de programação

- Variáveis, instruções condicionais (por exemplo, *if-then-else*, *case*), primitivas de iteração (por exemplo, *for*, *while*, *do-while*), funções
- Dicas:
  - Comentários, principalmente de variável
  - Uso constantes simbólicas: entrada tamanho, constante matemática, tamanho da estrutura de dados
  - Tipos enumerados

```
switch(cursuit) {
    case 'C':
      newcard.suit = C;
      break;
    case 'D':
      newcard.suit = D;
      break;
    case 'H':
      newcard.suit = H;
      break;
    case 'S':
      newcard.suit = S;
...
```

- Use subrotinas para evitar código redundante

```
...

    while (c != '0') {
        scanf("%c", &c);
        if (c == 'A') {
            if (row-1 >= 0) {
                temp = b[row-1][col];
                b[row-1][col] = ' ';
                b[row][col] = temp;
                row = row-1;
            }
        }
        else if (c == 'B') {
            if (row+1 <= BOARDSIZE-1) {
                temp = b[row+1][col];
                b[row+1][col] = ' ';
                b[row][col] = temp;
                row = row+1;
            }
        }

...
```

Entrada/Saída Padrão em C e C++

```c
#include<stdio.h>

int main() {
   long p,q,r;

   while (scanf("%ld %ld",&p,&q)
                !=EOF) {
     if (q>p) r=q-p;
     else r=p-q;

     printf("%ld\n",r);
   }
}
```

```cpp
#include<iostream.h>

void main()
{
long long a,b,c;

while (cin>>a>>b) {
   if (b>a)
      c=b-a;
   else
      c=a-b;
   cout << c << endl;
}
}
```

```
{$N+}
program acm;
var
  a, b, c : integer;
begin
  while not eof do
  begin
    readln(a, b);
    if b > a then
    begin
      c := b;
      b := a;
      a := c
    end;
    writeln(a - b);
  end
end.
```

Tipos de dados

- Vetor, Matriz, lista encadeada, árvore, grafo
- **Vetor**: armazena sequências de elementos de tipo único (inteiros, reais, registros)
  - Vetores de caracteres podem ser usados para representar strings
  - Uso de um valor **sentinela** (x)

```
i = n;
while ((a[i]>=x) && (i>=1)) {
        a[i] = a[i-1];
        i=i-1;
}
a[i+1] = x;
```

```
i = n;
a[0] = - MAXINT;
while (a[i] >= x) {
        a[i] = a[i-1];
        i=i-1;
}
a[i+1] = x;
```

Tipos de dados

- **Matrizes multidimensionais**: estruturas de grades retangulares
  - Exemplo: um vetor de n pontos no plano 2D pode ser pensado como um vetor $n \times 2$
    - índice j: 0 ou 1 de $A[i][j]$ determina se estamos referindo-se a coordenada x ou y do ponto
- **Registros**: pode agrupar registros de dados heterogêneos

```
struct ponto {
    int x, y;
};
```

  - Acessar valor: p.y e p.y

Referência

- SKIENA, Steven S; REVILLA, Miguel A. Programming challenges: the programming contest training manual. New York: Springer, 2003.