

Algoritmia e Programação

Introdução

Algoritmos e programas

Algoritmo

- Um algoritmo é uma especificação de uma sequência finita de operações que resolve um problema.
 - Finito : tem que resolver o problema, eventualmente.
 - Preciso : não pode ser ambíguo.
 - Efectivo : tem que resolver o problema, sempre.

Exemplo clássico: receita culinária



PANQUECAS TM31



Preparação:: 15min



3



Ingredientes

Imported

220 g de farinha
2 colheres de chá de fermento em pó royal
60 g de açúcar
2 ovos
1/4 de litro de leite
2 colheres de sopa de óleo de sésamo

Preparação

Colocar todos os ingredientes no copo e programar, 20 segundos, vel. 5.

Aquecer uma frigideira antiaderente, colocar uma colherada de massa, assim que começar a fazer bolhinhas por cima, virar com a ajuda de uma espátula para cozer do outro lado. O lume não deve estar muito forte para não queimarem.

Rende: 12 panquecas



Programa

- Implementação de um algoritmo em instruções compreendidas por um computador.
- Os computadores não improvisam.
 - O algoritmo tem que ser preciso e correcto!
 - Pode não ser... mas não vai funcionar... bem! :-D

Programa (exemplo)

```
#include <stdio.h>
```

```
main( )
```

```
{
```

```
    printf( "hello, world\n" );
```

```
}
```

Voltando aos algoritmos...

- Um algoritmo é habitualmente descrito através da composição de:
 - **dados do problema** : valores relevantes para o problema;
 - **operações** : acções que operam sobre os dados para produzir uma solução.

Estrutura de um algoritmo

- Nesta UC, vamos definir um algoritmo através de duas secções:
- ESTRUTURAS DE DADOS
 - Nesta primeira secção identificamos os dados do problema.
- ALGORITMO
 - Nesta segunda secção descrevemos a sequência de operações que conduzem à solução do problema.

Estrutura de um algoritmo

ESTRUTURAS DE DADOS:

Aqui definem-se os dados do problema.

ALGORITMO:

Aqui especificam-se as operações sobre os dados que conduzem à resolução do problema.

Variáveis

Dados de um problema

- Regra geral, os algoritmos definem operações que manipulam e processam **dados**.
- Os dados são valores que definem a instância de um tipo de problema.
 - Exemplos: velocidade, aceleração, temperatura, saldo bancário, número de pessoas, etc.
- Esses dados têm que ser representados na definição de um algoritmo.

Dados de um problema

- Os dados podem ser de dois tipos:
 - **constantes:** valores que são sempre iguais em todas as instâncias de um problema.
 - Exemplos: velocidade da luz no vazio, constante de Plank.
 - **variáveis:** valores que mudam em cada instância do problema, e até durante uma instância do problema.
 - Exemplos: número de pessoas no cinema, peso transportado por um veículo, saldo bancário.

Variáveis e constantes

- Num algoritmo...
 - um dado constante pode ser representado pelo seu valor, ou preferencialmente por um nome: a **constante**.
 - um dado variável tem que ser representado de forma abstracta por um nome: a **variável**.

Definir uma variável

- Na definição de um algoritmo, é habitual indicar o **tipo** de valores que uma variável pode representar. Os mais básicos são:
 - **INTEIRO** : valores inteiros, sem parte fraccionária.
 - **REAL** : valores com parte fraccionária.
 - **TEXTO** : mensagens textuais.
- Para além do tipo, escolhe-se um **nome** que deverá ser facilmente associado ao valor representado pela variável.

Definir variáveis e constantes (exemplo)

ESTRUTURAS DE DADOS:

INTEIRO numero_medicoes, temperatura

REAL media_temperatura

TEXT0 mensagem_sensor

CONSTANTE $PI = 3.1415$

Atribuir um valor a uma variável

- Uma variável armazena **um único** valor, em cada instante.
- O valor é atribuído com o operador de atribuição: = (símbolo “igual”).
- Atribuir um novo valor a uma variável **destrói** o valor anteriormente armazenado!

Atribuir um valor a uma variável (exemplo)

ESTRUTURAS DE DADOS:

INTEIRO: temperatura_celsius, temperatura_fahrenheit

ALGORITMO:

temperatura_celsius = 23

(...)

temperatura_fahrenheit = temperatura_celsius * 1.80 + 32

Entrada e saída

Operações de entrada e saída

- Muitos algoritmos requerem interacção com o exterior.
 - Receber dados do exterior: operação de **entrada**.
 - Enviar dados para o exterior: operação de **saída**.

Operação de entrada

- Indica que se deve ler um valor para uma variável.

ESTRUTURAS DE DADOS:

INTEIRO: temperatura_celsius

ALGORITMO:

(...)

LER(temperatura_celsius)

Operação de saída

- Indica que se deve apresentar dados.
 - Pode ser uma variável ou uma mensagem.

ESTRUTURAS DE DADOS:

INTEIRO: temperatura_celsius

ALGORITMO:

ESCREVER("Bom dia!")

(...)

ESCREVER(temperatura_celsius)

Exemplo (juntando tudo)

ESTRUTURAS DE DADOS:

INTEIRO: temperatura_celsius, temperatura_fahrenheit

ALGORITMO:

LER(temperatura_celsius)

$\text{temperatura_fahrenheit} = \text{temperatura_celsius} * 1.80 + 32$

ESCREVER(temperatura_fahrenheit)

FIM

Traçagem

Traçagem de um algoritmo

- A traçagem é um processo em que se analisa o comportamento de um algoritmo, avaliando a sua execução passo a passo.
- Para algoritmos que recebem valores do exterior, é estabelecido para cada traçagem um conjunto de valores de entrada, a que corresponderá um determinado resultado esperado.
- A traçagem, permite detectar erros conceptuais no algoritmo, mas não permite certificar que o algoritmo está isento de erros!

Exemplo de traçagem

- Executar a traçagem do algoritmo com os seguintes valores de entrada:
 - temperatura_celsius : 0

| Algoritmo | Traçagem |
|--|-----------------------------|
| LER(temperatura_celsius) | temperatura_celsius = 0 |
| temperatura_fahrenheit = temperatura_celsius * 1.80 + 32 | temperatura_fahrenheit = 32 |
| ESCREVER(temperatura_fahrenheit) | SAIDA: 32 |

Implementação em C

Implementação do exemplo em C

```
#include <stdio.h>
```

```
int main() {  
    int temperatura_celsius, temperatura_fahrenheit;  
  
    printf("Temperatura em Celsius: ");  
    scanf("%d", &temperatura_celsius);  
  
    temperatura_fahrenheit = temperatura_celsius * 1.8 + 32;  
  
    printf("Temperatura: %d Fahrenheit\n", temperatura_fahrenheit);  
  
    return 0;  
}
```

Implementação do exemplo em C

```
#include <stdio.h>
int main() {
    int temperatura_celsius, temperatura_fahrenheit;

    printf("Temperatura em Celsius: ");
    scanf("%d", &temperatura_celsius);

    temperatura_fahrenheit = temperatura_celsius * 1.8 + 32;

    printf("Temperatura: %d Fahrenheit\n", temperatura_fahrenheit);

    return 0;
}
```

Incluir funções de entrada/saída.

Declaração das variáveis.
Valores iniciais indefinidos.

Função que arranca o programa.

Ler valor inteiro decimal (com sinal)
para (&) a variável temperatura_celsius.

Apresentar valor inteiro decimal (com sinal)
da variável temperatura_fahrenheit.

Termina main() com sucesso.