

Aula 06

MC 102 - Algoritmos e Programação de Computadores

Comandos Condicionais II Biblioteca math.h

2o. Sem 2007

Algoritmos e Programação de Computadores - Turmas I J K L

1

Teste de 1 RA

Dado um RA, verificar se o aluno encontra-se matriculado em MC102.

```
int main () {
    int ra;
    scanf("%d", &ra);
    if (ra == 070101) {
        printf("O aluno %d esta matriculado em MC102\n", ra);
    }
}
```

2o. Sem 2007

Algoritmos e Programação de Computadores - Turmas I J K L

3

Decisão Múltipla

O que fazer quando temos que verificar diversos valores para uma mesma variável?

Por exemplo, dado um RA, verificar se o aluno encontra-se matriculado em MC102.

2o. Sem 2007

Algoritmos e Programação de Computadores - Turmas I J K L

2

Teste de 2 RAs

Dado dois RA, verificar se ambos alunos encontram-se matriculados em MC102.

```
int main () {
    int ra;
    scanf("%d", &ra);
    if (ra == 070101 || ra == 070103){
        printf("O aluno %d esta matriculado em MC102\n", ra);
    }
}
```

2o. Sem 2007

Algoritmos e Programação de Computadores - Turmas I J K L

4

Teste de n RAs

Problema: cada turma de MC102 possui cerca de 60 alunos e temos muitas turmas neste semestre.

```
if (ra == 2582 || ra == 10129 || ra == 16267 || ...
ra == 962185) {
    printf("O aluno %d esta matriculado em MC102.\n", ra);
}
else { ...
```

Teste de n RAs

Problema: cada turma de MC102 possui cerca de 60 alunos e temos muitas turmas neste semestre.

```
if (ra == 2582 || ra == 10129 || ra == 16267 || ...
ra == 962185) {
    printf("O aluno %d esta matriculado em MC102.\n", ra);
}
else { ...
```

Muitas condições a serem testadas!

Decisão Múltipla – Caso 2

Dado um RA, escrever o nome do aluno

```
int main () {
    int ra;

    scanf("%d", &ra);

    if (ra == 10129)
        printf("Maria das Dores\n");
    if (ra == 33860)
        printf("João Pedro\n");
    if (ra == 33967)
        printf("Leonardo Camargo\n");
}
```

Decisão Múltipla – Caso 2

Dado um RA, escrever o nome do aluno

```
int main () {
    int ra;

    scanf("%d", &ra);

    if (ra == 10129)
        printf("Maria das Dores\n");
    if (ra == 33860)
        printf("João Pedro\n");
    if (ra == 33967)
        printf("Leonardo Camargo\n");
}
```

É possível diminuir o num. de testes?

Decisão Múltipla – Caso 2

Uso de if ... else

```
int main () {
    int ra;

    scanf("%d", &ra);
    if (ra == 10129)
        printf("Maria das Dores\n");
    else if (ra == 33860)
        printf("João Pedro\n");
    else if (ra == 33967)
        printf("Leonardo Camargo\n");
}
```

Decisão Múltipla – Caso 2

Uso de if ... else

```
int main () {
    int ra;

    scanf("%d", &ra);
    if (ra == 10129)
        printf("Maria das Dores\n");
    else if (ra == 33860)
        printf("João Pedro\n");
    else if (ra == 33967)
        printf("Leonardo Camargo\n");
}
```

Novamente vários testes!

Comando switch

Simplifica uma expressão onde uma variável inteira (ou caracter) deve realizar diferentes operações dependendo exclusivamente de seu valor.

```
switch (variável) {
    case valor_1:
        comandos;
        break;
    case valor_2:
        comandos;
        break;
    default:
        comandos;
}
```

Comando switch

Simplifica uma expressão onde uma variável inteira (ou caracter) deve realizar diferentes operações dependendo exclusivamente de seu valor.

```
switch (variável) {
    case valor_1:
        comandos;
        break;
    case valor_2:
        comandos;
        break;
    default:
        comandos;
}
```

Comandos a serem executados vêm após : de cada valor testado

Comando switch

Simplifica uma expressão onde uma variável inteira (ou caracter) deve realizar diferentes operações dependendo exclusivamente de seu valor.

```
switch (variável) {
```

```
    case valor_1:
        comandos;
        break;
```

```
    case valor_2:
        comandos;
        break;
```

```
    default:
        comandos;
```

```
}
```

Executa os comandos até encontrar um **break** ou até o fim do switch

Comando switch

Simplifica uma expressão onde uma variável inteira (ou caracter) deve realizar diferentes operações dependendo exclusivamente de seu valor.

```
switch (variável) {
```

```
    case valor_1:
        comandos;
        break;
```

```
    case valor_2:
        comandos;
        break;
```

```
    default:
        comandos;
```

```
}
```

Os comandos da opção **default** serão executados caso não encontre algum case correspondente ao valor sendo testado

Comando switch

Um conjunto de valores pode ser verificado em sequência.

```
switch (variável) {
    case valor_1:
    case valor_2:
    case valor_3:
    case valor_4:
    ...
    case valor_n:
        comandos;
        break;
}
```

```
switch (variável) {
    case valor_1:
    case valor_2:
        comandos;
        break;
    case valor_3:
    case valor_4:
        comandos;
        break;
}
```

switch – Caso 1

Dado um RA, verificar se o aluno encontra-se matriculado na turma

```
switch (ra) {
    case 070101:
    case 070103:
    case 070105:
    case 070107:
    ...
        printf("Aluno matriculado em MC102.\n");
        break;
    default:
        printf("Aluno não matriculado em MC102.\n");
}
```

switch – Caso 2

Dado um RA, escrever o nome do aluno

```
switch (ra) {
    case 070101:
        printf("José de Oliveira.\n");
        break;
    case 070103:
        printf("Maria dos Santos.\n");
        break;
    case ...
    default:
        printf("Aluno não matriculado.\n");
}
```

Programa 1

Faça um programa que verifique se um caracter digitado é numérico (0 a 9) e, caso contrário, se é uma vogal ou uma consoante, ou outro caracter qualquer.

Programa 1 - Solução

```
#include <stdio.h>

int main(int argc, char **argv) {
    char c;
    printf("Entre com uma tecla:");
    scanf("%c", &c);
    switch (c) {
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            printf("\nVoce digitou um caracter numerico!\n");
            break;
        case 'A': case 'E': case 'I': case 'O': case 'U':
        case 'a': case 'e': case 'i': case 'o': case 'u':
            printf("\nVoce digitou uma vogal!\n");
            break;
        case 'B': case 'C': case 'D': case 'F': case 'G':
        case 'H': case 'J': case 'K': case 'L': case 'M':
        (...):
            printf("\nCaracter não é numérico, vogal ou consoante!\n");
            break;
    }
```

Programa 1 – Solução (cont.)

```
(...)
    case 'N': case 'P': case 'Q': case 'R': case 'S':
    case 'T': case 'V': case 'W': case 'X': case 'Y':
    case 'Z': case 'b': case 'c': case 'd': case 'f':
    case 'g': case 'h': case 'j': case 'k': case 'l':
    case 'm': case 'n': case 'p': case 'q': case 'r':
    case 's': case 't': case 'v': case 'w': case 'x':
    case 'y': case 'z':
        printf("\nVoce digitou uma consoante!\n");
        break;
    default:
        printf("\nA tecla digitada não é numérica, ");
        printf("nem uma vogal e nem uma consoante!\n");
}
```

Exercício 1

Faça um programa que dados um mês e um ano (anterior a 2011) ele calcula o número de dias deste mês.

Biblioteca math.h

Biblioteca que inclui funções e constantes matemáticas

Como deixá-la disponível no programa?

No cabeçalho do programa:

```
#include <math.h>
```

Biblioteca math.h

Lembrem-se!

Para compilar no linux, use:

```
gcc meu_prog.c -o meu_prog -lm
```

Biblioteca math.h

Algumas constantes úteis:

M_E	e
M_LOG2E	$\log_2 e$
M_LOG10E	$\log_{10} e$
M_LN2	$\log_e 2 = \ln 2$
M_SQRT2	$\sqrt{2}$
M_PI	π

Atenção: algumas delas não estão mais definidas no C99

Biblioteca math.h

Algumas funções úteis:

double ceil (double x)	<i>menor inteiro maior que x</i>
double floor (double x)	<i>maior inteiro menor que x</i>
double fabs (double x)	<i>valor absoluto de x</i>
double log (double x)	$\ln x$
double log10 (double x)	$\log x$
double pow (double x, double y)	x^y
double exp (double x)	e^x
double sqrt (double x)	$\sqrt{x}$
double sin (double x)	$\sin x$
double cos (double x)	$\cos x$
double tan (double x)	$\tan x$

Exercício 2

Faça um programa que leia dois número inteiros (a e b) e um operador (+ p/ adição, - p/ subtração, * para multiplicação, / para divisão e ^ para potência) e realize a operação **a <op> b**.

Exercício 3

Escreva um programa que lê os valores de **a**, **b** e **c** e encontra as raízes reais da equação $ax^2+bx+c=0$.

Exercício 2- Solução

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

int main(int argc, char **argv) {
    char c;
    int a, b;
    float res;

    printf("Entre com dois inteiros e um operador (+ - * / ^):");
    scanf("%d %d %c", &a, &b, &c);
    switch (c) {
        case '+':
            res = a + b; break;
        case '-':
            res = a - b; break;
        case '*':
            res = a * b; break;
        case '^':
            res = pow(a, b); break;
    }
    printf("Resultado: %f\n", res);
}
```

Exercício 2 – Solução (cont.)

(...)

```
case '/':
    res = a / b; break;
case '^':
    res = pow(a, b); break;
default:
    printf("Operador '%c' invalido!\n", c);
}
switch (c) {
case '+':
case '-':
case '*':
case '/':
case '^':
    printf("\n%d%c%d = %f", a, c, b, res);
```