

LABORATÓRIO 22

INSTRUÇÃO DE DESVIO SWITCH

EXERCÍCIOS DE REVISÃO

VOCÊ DEVE ACOMPANHAR PARA OBTER INFORMAÇÕES COMPLEMENTARES

1. Construa uma expressão condicional para obter o valor absoluto de uma variável. Isto é, se a variável x é positiva, o valor da expressão deve ser x , mas se x é negativa, o valor da expressão deve ser $-x$, tornando a expressão positiva.

Digite um número positivo ou negativo: -5
O valor absoluto de -5 é 5.

2. No programa abaixo, quais as vantagens de usar o tipo caractere para a variável **escolha** e as constantes 'a', 'b', 'c', 'd' para os rótulos do switch, no lugar de usar números para as escolhas do menu?

```
#include <iostream>
using namespace std;
int main()
{
    int escolha;
    cin >> escolha;
    while (escolha != 5) {
        switch (escolha) {
            case 1 : cout << "Boa escolha\n."; break;
            case 2 : cout << "Estou na dúvida\n."; break;
            case 3 : cout << "Não sei o que dizer.\n"; break;
            case 4 : cout << "Quatro é um número\n."; break;
            default: cout << "Esta não é uma opção.\n";
        }
        cin >> escolha;
    }

    return 0;
}
```

Dica: pense no que aconteceria se o usuário digitasse 5 em cada caso e o que aconteceria se o usuário digitasse 's' em cada caso.

EXERCÍCIOS DE FIXAÇÃO

VOCÊ DEVE FAZER OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Reescreva o programa abaixo de forma que ele não utilize as instruções break e continue, e ainda assim execute a mesma função.

```
#include <iostream>
#include <cctype>
using namespace std;
int main()
{
    int letras = 0;
    char ch;
    cin >> ch;
    while (ch != '!')
    {
        if (ch == '?')
            break;
        if (!isalpha(ch))
        {
            cin >> ch;
            continue;
        }
        letras++;
        cin >> ch;
    }

    return 0;
}
```

2. Escreva um programa que aceite entrada de texto até achar o caractere @ e que exiba os caracteres, exceto os dígitos, convertendo maiúsculas para minúsculas e vice-versa.

Exemplo 1:

```
Digite seu texto (@ para sair):
hoje vamos programar muito @ mais que ontem
HOJE VAMOS PROGRAMAR MUITO
```

Exemplo 2:

```
Digite seu texto (@ para sair):
Os 12 bravos homens lutaram
oS BRAVOS HOMENS LUTARAM
ate a morte naqueles 15 Minutos de Batalha. @
ATE A MORTE NAQUELES mINUTOS DE bATALHA.
```

3. Em uma empresa, um funcionário pode ser conhecido pelo seu nome, seu cargo ou seu apelido. Escreva um programa que possa listar funcionários por nome, cargo, apelido ou pela preferência do funcionário. Utilize o registro abaixo como base:

```
struct bop {  
    char nome[50];    // nome do funcionário  
    char cargo[50];   // cargo que ocupa na empresa  
    char apelido[50]; // apelido do funcionário  
    int pref;         // 0 = nome, 1 = cargo, 2 = apelido  
}
```

No programa crie um pequeno vetor desses registros e inicialize-o para valores de sua escolha. Faça o programa rodar um laço que possibilite o usuário selecionar uma das alternativas:

- a. Mostrar por nome
- b. Mostrar por cargo
- c. Mostrar por apelido
- d. Mostrar por preferência
- e. Sair

Observe que “Mostrar por preferência” não significa mostrar o campo pref do registro; significa mostrar o campo correspondente ao número armazenado em pref. Por exemplo, se a preferência de João é 1 e de José é 2, então escolhendo a opção d, exibiria o cargo de João e o apelido de José. O programa deve rodar semelhante ao exemplo abaixo:

Listagem de Funcionários

- a. Mostrar por nome
- b. Mostrar por cargo
- c. Mostrar por apelido
- d. Mostrar por preferência
- e. Sair

Digite sua opção: a

José
Marcos
João
Carlos

Digite sua opção: d

Paquito
Porteiro
Segurança
Carlos

Digite sua opção: e

Tchau!

4. Escreva um programa que leia a entrada do usuário palavra por palavra até que seja achada uma palavra apenas com a letra 's'. O programa deve então relatar o número de palavras que começam por vogais, o número de palavras que começam por consoantes, e o número de palavras que não se encaixam nos casos anteriores. A saída do programa deve ser semelhante ao exemplo abaixo:

```
Digite palavras (s para sair):  
Os 12 bravos homens lutaram  
ate a morte naqueles 15 minutos de batalha. s  
3 palavras começam por vogais  
8 palavras começam por consoantes  
2 outras palavras
```

Dica: uma solução é usar `isalpha()` para discriminar entre palavras que começam com letras e aquelas que não começam e então usar um `if` ou `switch` para identificar aquelas que começam com vogal.

5. Escreva um programa que apresente um menu com 4 opções, cada uma rotulada por uma letra. Se o usuário responder com uma letra diferente das quatro opções, o programa deve pedir ao usuário que entre com uma opção válida, até que ele o faça. O programa deve usar um `switch` para selecionar e exibir uma frase em resposta a uma escolha válida.

```
Menu do Sistema  
  
a) Inserir  
b) Remover  
c) Buscar  
d) Sair  
  
Escolha: f  
Opção Inválida  
  
Escolha: g  
Opção Inválida  
  
Escolha: 4  
Opção Inválida  
  
Escolha: a  
Inserindo elementos...  
  
Escolha: b  
Removendo elementos...  
  
Escolha: c  
Buscando elementos...  
  
Escolha: d  
Tchau!
```

Dica: observe que o programa funciona corretamente mesmo se forem digitados números no lugar de letras.

EXERCÍCIOS DE APRENDIZAGEM

VOCÊ DEVE ESCREVER PROGRAMAS PARA REALMENTE APRENDER

1. O supermercado on-line ABC vende alface por R\$1,25/Kg, beterraba por R\$0,65/Kg e cenoura por R\$0,90/Kg. Ele dá um desconto de 5% para pedidos que ultrapassem R\$100 (valor calculado sem os custos de envio da mercadoria). O custo de entrega é de R\$3,50 para pedidos com menos de 5Kg, R\$10,00 para pedidos de 5Kg até 20Kg, e R\$8,00 mais R\$0,10 por Kg para pedidos iguais ou maiores que 20Kg.

Escreva um programa que permita ao cliente fazer um pedido de compra de vegetais. O algoritmo deve fornecer um menu com os vegetais disponíveis e os respectivos preços:

Supermercado ABC

a) Alface R\$ 1,25/Kg
b) Beterraba R\$ 0,65/Kg
c) Cenoura R\$ 0,90/Kg

Faça seu pedido usando as letras (a, b, c) e (s) para encerrar:

Ele deve então ler uma opção até que o cliente encerre o pedido: uma resposta "a" deve permitir ao usuário entrar com os quilos de alface, "b" com os quilos de beterraba, "c" com os quilos de cenoura e "s" deve encerrar o pedido. O algoritmo deve permitir que o usuário faça a correção do pedido quantas vezes ele quiser antes de encerrar o pedido. Se, por exemplo, ele pressionar a letra "a" uma segunda vez, considere que a segunda entrada é uma correção da quantidade de alface digitada anteriormente. O pedido só deve ser encerrado quando o usuário digitar "s", até lá o algoritmo deve aceitar pedidos e correções de pedidos indefinidamente.

Quando o usuário encerrar o pedido, o algoritmo deve mostrar o total do pedido, o desconto (se tiver um), o custo da entrega e o total a pagar (incluindo todos os custos e descontos).

Produto	Preço/Kg	Pedido (Kg)	Total Parcial
-----	-----	-----	-----
Alface	R\$ 1,25/Kg	3,2Kg	R\$ 4,00
Beterraba	R\$ 0,65/Kg	6,0Kg	R\$ 3,90
Cenoura	R\$ 0,90/Kg	10Kg	R\$ 9,00
-----	-----	-----	-----

Se o usuário fizesse o pedido acima, a saída seria:

Total do pedido:	R\$ 16,90
Descontos:	R\$ 0,00
Custo de entrega:	R\$ 10,00
-----	-----
Total a pagar:	R\$ 26,90

2. Escreva um programa que aceite entrada de texto até achar o caractere @. O programa deve então usar uma instrução switch para contar a quantidade de cada vogal existente no texto. O texto pode conter letras, números, espaços e caracteres de pontuação.

```
Digite uma frase (@ para finalizar):
Ontem eu andei sob chuva forte, eram 12 em ponto.@
a:3
e:6
i:1
o:5
u:2
```

3. Crie um registro “Prato” com os campos nome e preço. Na função principal, crie um vetor de pratos com os pratos de uma lanchonete, inicializando-o com pratos de sua escolha. Mostre o cardápio dos pratos da lanchonete para o usuário e pergunte se ele quer pedir algo. Enquanto ele fizer pedidos, some o valor do prato pedido ao valor total da conta que ele vai pagar, e mostre o cardápio novamente. Quando ele quiser encerrar o pedido, mostre a conta final com o total a pagar.
4. Faça um programa que receba o preço de um produto e seu código de origem e mostre sua procedência. A procedência obedece a tabela a seguir:

Código de Origem	Procedência
1 ou 2	Sul
3	Norte
4	Centro-oeste
5 ou 6	Nordeste
7 ou 8 ou 9	Sudeste

Se o código digitado pelo usuário não for nenhum dos especificados acima, mostre a mensagem "Código Inválido".

5. Crie um programa que simule uma minicalculadora. Está minicalculadora deve receber dois operandos e operá-los de acordo com o operador escolhido. Os operadores disponíveis devem ser os de adição, subtração, multiplicação e divisão, e a minicalculadora deve ser capaz de trabalhar tanto com inteiros como com decimais. O usuário deve ser capaz de escolher qual operação realizar e depois inserir os operandos. Ele também deve ser capaz de realizar quantas operações quiser até que deseje parar.