1. Explique o funcionamento do mecanismo chamado “call-back function”.

Resposta:

É uma função que é passada como argumento por outra função, como por exemplo, a função WindowProcedure do main.cpp, em que é passada como argumento para a função DispatchMessage.

2. Explique o funcionamento do código contido no arquivo “main.cpp”.

Resposta:

O código gera um programa com interface e janela própria.

3. O que o código contido no arquivo “main1.cpp” faz?

Resposta:

Além de imprimir o “Hello Word”, ele tem um loop que transforma caracteres minúsculos em maiúsculo.

4. Como este código funciona?

Resposta:

Primeiramente ele imprime a string “Hello Word”, em seguida ele entra em um laço que pega todos os caracteres e diminui seu valor inteiro em -32, contudo o laço é quebrado se o usuário colocar o caractere “s” .

5. Por que é necessário utilizar as instruções de “type casting”?

Resposta:

Pois precisamos transformar o valor char em int para fazer a manipulação do tipo de caractere, para percorrer a tabela ASCII e em seguida converter novamente para char.

6. O que significa “0xFF” no código “main2.cpp”?

Resposta:

O “0x??” representa inteiros hexadecimais

7. O que o primeiro loop do código “main2.cpp” faz e como ele funciona?

Resposta:

O primeiro loop imprime a string “Tecnicas de Programa��o II” usando o deslocamento do endereço do ponteiro, até que o ponteiro aponte para uma quebra de linha

8. O que o segundo loop do código “main2.cpp” faz e como ele funciona?

Resposta:

O segundo loop imprime o vetor de inteiros, usando deslocamento de ponteiro, e cada loop ele imprime o número apontado pelo ponteiro e o endereço que ele está apontando.

9. O que o terceiro loop do código “main2.cpp” faz? Por que ele não funciona?

Reposta:

No terceiro loop, ele transforma o vetor de números para um ponteiro de char, contudo ele não aponta para nada, logo o loop não imprime nem o conteúdo e nem o endereço.

10. O que pode ser concluído sobre a aritmética de ponteiros?

Reposta:

Que apesar de complicada e sensível, pois o mau uso pode comprometer a boa execução do código, seu uso é essencial e faz o diferencial da linguagem c e suas derivações.

11. O código possui duas diretivas de pré-compilação que fazem com que o código “main3.cpp” gere erros. Compile e execute o código com a diretiva ERRO1 ativada e desativada e, então, Bbaseado na saída impressa na tela, explique o que o trecho de código de linha 27 a 32 faz.

Resposta:

Primeiramente ele atribui um endereço para um ponteiro, em seguida imprime o ponteiro, o endereço e o valor correspondente, em seguida ele imprime a variável, que altera dependendo se é estático ou não, ou seja varia se o ERRO1 está definido ou não.

12. O código possui duas diretivas de pré-compilação que fazem com que o código “main3.cpp” gere erros. Compile e execute o código com a diretiva ERRO2 ativada e desativada e, então, baseado na saída impressa na tela, explique o funcionamento do trecho de código entre as linhas 37 e 57.

Resposta:

Primeiramente ele vai atribuir um valor nulo ou a variável. logo se não for alocado um valor ao ponteiro e tentar imprimi lo, vai gerar uma falha de segmentação, isso irá ocorrer se o ERRO2 for definido.