

Lista 7 – estrutura de controle: repetição (sala de aula)

Assunto:

Variáveis, instruções de entrada e saída de dados e estruturas de decisão e de repetição, enfatizando validação de entradas e execução repetida do programa.

Forma de resolução:

- a) Se necessário:
 - a.1) representar o algoritmo em descrição narrativa, utilizando o seguinte procedimento para análise do problema e definição da solução:
 - a.1.1) Dados de entrada
 - a.1.2) Processamento (instruções a serem realizadas)
 - a.1.3) Dados de saída
 - a.2) representar o algoritmo em fluxograma.
- b) Representar o algoritmo de solução do problema na linguagem C, com as entradas informadas pelo usuário.

Importante:

- a) Chaves
 - a. Elas delimitam um bloco de instruções (comandos) que estão subordinados a uma cláusula if, else ou else if ou a uma estrutura de repetição do, while e do while.
 - b. As instruções delimitadas por um conjunto de chaves deve obrigatoriamente estar indentada. Um conjunto de espaços (tab) de avanço.
 - c. Cada chave aberta deve ter a sua correspondente fechada. Para facilitar, sempre, abra uma chave, feche-a e retorne para escrever o código (as instruções) dentro dessas chaves.
 - d. Para haver um teste lógico é obrigatória a existência de if e vice-versa. O teste lógico é colocado dentro de parênteses.
- b) Erros:
 - a. Verificar se as chaves estão corretas.
 - b. Verificar se há teste lógico sem if.
 - c. Loop infinito. A variável de controle não está sendo incrementada ou decrementada, o teste lógico que verifica a condição de continuidade ou parada não está sendo realizado corretamente.

Exercícios:

Observações: Para todos os exercícios desta lista:

a) fazer os programas de forma que o usuário possa optar por repetir a execução dos mesmos. Ao final de uma execução questionar se o usuário quer continuar utilizando o programa. Se informado afirmativamente, repetir a execução. Isso pode ser feito utilizando uma estrutura do while que conterá a parte do programa que será repetida. As variáveis que não dependem de dados de entrada ou de execução podem ser declaradas fora dessa estrutura de repetição do programa. Atenção para a inicialização de variáveis: deve ser feita dentro dessa estrutura de repetição.

b) Validar as entradas.

c) Validar para que não sejam realizadas divisões por zero.

1) Um determinado material radioativo perde metade de sua massa a cada 500 segundos. Dada a massa inicial, em gramas, fazer um programa que determine o tempo necessário para que essa massa se torne menor que 0,05 gramas. Escreva a massa inicial, a massa final e o tempo calculado em horas, minutos e segundos. A massa inicial informada pelo usuário tem que ser maior que zero. Validar essa entrada, isto é, ficar repetindo a leitura da massa inicial até que o usuário informe um valor válido, ou seja, maior que zero.

2) Escreva um programa que imprima todas as possibilidades de que no lançamento de dois dados a soma das faces seja um valor informado pelo usuário. O usuário deve informar um valor válido, entre 2 a 12. Repetir a leitura até que o usuário informe um valor válido. Mostrar da seguinte forma, para, por exemplo, informado o valor 7:

$$1 + 6 = 7$$

$$2 + 5 = 7$$

$$3 + 4 = 7$$

$$4 + 3 = 7$$

$$5 + 2 = 7$$

$$6 + 1 = 7$$

3) Ler um número N qualquer maior ou igual a 1 e menor ou igual a 50 e apresentar o valor obtido da multiplicação sucessiva de N por 2, enquanto o produto for menor que 250 ($N*2$; $N*2*2$; $N*2*2*2$; etc.). O valor N deverá ser verificado quanto a sua validade. Caso o usuário informe um valor fora da faixa, o programa deverá informar que o valor digitado não é válido e repetir a leitura.

Atenção: o produto deve ser menor que 250.

4) Escreva um programa que leia o valor de uma aplicação, o percentual de rendimento mensal obtido por esta aplicação e o período do investimento; e retorne o valor da aplicação ao final do período de investimento. Obs.: a cada 12 meses o percentual de rendimento deve ser aumentado em 0,25%.

5) Apresentar os números primos entre 1 e 100.

6) Apresentar os n primeiros números pares e divisíveis por 3 informados pelo usuário. Se o usuário informar um número negativo transformá-lo em positivo antes de utilizar.

Por exemplo:

Informe a quantidade de números pares a serem mostrados:

3

mostrar da seguinte forma (incluindo a mensagem antes dos números):

Os 3 primeiros números pares são: 6 12 18

7) No intervalo entre 0 e 100, utilizando uma estrutura for para mostrar cada um dos itens em listagens separadas. Colocar um cabeçalho/título que identifique cada uma das listagens:

- a) os valores desse intervalo em ordem decrescente.
- b) os valores pares desse intervalo, sem utilizar if, apenas a variável de inicialização, controle e incremento/decremento do for.
- c) os múltiplos de 10, em ordem crescente e sem utilizar if, apenas a variável de inicialização, controle e incremento/decremento do for.
- d) os múltiplos de 5, em ordem decrescente e sem utilizar if, apenas a variável de inicialização, controle e incremento/decremento do for.