1. Um programa em C é basicamente estruturado em blocos de código. Blocos nada mais são que conjuntos de instruções, e devem ser delimitados com chaves ({...}). Um bloco também pode conter outros blocos. Desta forma, analise o trecho de código C a seguir.

1 int a;

2 {

3 int a;

4 int b;

5 {

6 int c;

7 }

8 }

9 {

10 int b;

11 }

Analisando o trecho de código acima, assinale a opção que possui o número da linha onde está localizado a variável com escopo mais global.

1. 3
2. 1
3. 6
4. 10
5. 4
6. Nas linguagens de programação, os nomes dados às variáveis, funções e rótulos são chamados de identificadores. Um programador deve seguir algumas regras para nomear seus identificadores sob pena se seu programa não compilar, caso não o faça corretamente.

Tendo como base as regras de nomenclatura de identificadores da linguagem C, analise os identificadores nas alternativas a seguir.

1. 1contador
2. Casa2
3. Oi!
4. Conta\_corrente

Estão corretos apenas os identificadores apresentados em:

1. 1 e 2
2. 2 e 3
3. 1 e 3
4. 2 e 4
5. 1, 2 e 4
6. Ao declarar uma variável para armazenamento de dados na  
   implementação de um programa, deve-se especificar o seu  
   tipo. Analise as premissas abaixo sobre tipos de dados.

I - 0 tipo de dado "float" serve para armazenar números de  
ponto flutuante, ou seja, com casas decimais. O padrão mais  
utilizado nos últimos 10 anos é o IEEE 754-1985.  
Desta forma, analise o trecho de código em C a seguir.

Il - O tipo de dado double serve para armazenar números de  
ponto flutuante de dupla precisão, normalmente tem o dobro do  
tamanho do "float" e, portanto o dobro da capacidade. O padrão  
mais adotado também é o IEEE 754-1985.

III - O tipo char ocupa 1 byte, e serve para armazenar  
caracteres ou inteiros. Isso significa que o programa reserva  
um espaço de 8 bits na memória RAM ou em registradores do  
processador para armazenar um valor.

IV - Com vetores do tipo char é possível criar cadeias de  
caracteres (strings).

Está correto o que se afirmar em:

a) I, II, IIl e IV.

b) IIl e IV, apenas.

c) I, Il e IV, apenas.

d) IV, apenas.

e) l e ll, apenas.

1. Lógica de Programação é a técnica de desenvolver algoritmos  
   (sequências lógicas) para atingir determinados objetivos dentro  
   de certas regras baseadas na Lógica matemática e em outras  
   teorias básicas da Ciência da Computação e que depois são  
   adaptados para a Linguagem de Programação utilizada pelo  
   programador para construir seu software.

Avalie as expressões a seguir e correlacione as colunas,  
classificando o resultado por meio os seguintes códigos:

I. Array Caractere. II. Lógica. III. Inteiro IV. Real.

( )126

( ) 3<8

( ) "32657"

( ) 6.548

Assinale a opção que corresponde à sequência correta das  
colunas anteriores.

a) l, lI, IV e III.

b) IlI, IV, II e I.

c) lI, I, III e IV.

d) III, Il, I e IV.

e) II, I, IV e III.

1. Analise a utilização de ponteiros no seguinte trecho de código,  
   escrito na linguagem C.

int x, \*p1, \*\*p2;

x = 100;

p1 = &x;

p2 = &p1;

printf("%d", \*\*p2);

Assinale a opção que contém o resultado da execução do  
código apresentado.

a) O valor correspondente ao endereço de p1 é exibido na tela.

b) O valor 100 é exibido na tela.

c) O valor correspondente ao endereço de x é exibido na tela.

d) Ocorre um erro na execução da linha do comando printf.

e) O código não compila.

1. Analise o seguinte trecho de código em linguagem C.

trocou=1;  
while ( trocou == 1 ) {

trocou=0;  
for (i=0;i<9;i++) {

if(vet[i] > vetſi +1]) {

aux = vet[i];

vet[i] = vetți +1];

vet[i+1] = aux;

trocou=1; }

}

}

Analise as assertivas a respeito da execução do código acima.

1. Ordena os 10 números do vetor em ordem crescente.
2. Ordena os 10 números do vetor em ordem decrescente.
3. Determina o maior valor presente no vetor.
4. Representa o algoritmo de ordenação denominado bolha.

Está correto o que se afirma apenas em:

1. III
2. I
3. II e IV
4. IV
5. I e IV
6. Um ponteiro é um tipo de dado de uma linguagem de  
   programação cujo valor refere-se diretamente a um outro valor  
   alocado em outra área da memória.

Desta forma, analise o trecho de código a seguir.

int \*pti;

int i = 10;

pti = &i;

Assinale a opção que apresenta uma análise **falsa** sobre o  
trecho do código acima.

a) pti é igual a 10.

b) \*pti é igual a 10.

c) Ao se alterar o valor de i, \*pti será modificado.

d) pti armazena o endereço de i.

e) Ao se executar \*pti = 20; i passará a ter o valor 20.

1. Arquivo é uma das maneiras mais importantes e utilizadas em  
   C para salvar informações. A forma de acesso a este arquivo é  
   feito em modos.

Desta forma, assinale a opção que possui uma descrição  
**incorreta** sobre os modos de acesso a arquivos em C.

a) O modo **r** abre o arquivo somente para leitura e criará um arquivo novo se não existir.

b) O modo **a+** abre o arquivo para escrita no final do arquivo e leitura.

c) O modo **w** abre o arquivo somente para escrita no início do arquivo. Apagará o conteúdo do arquivo se ele já existir, criará um arquivo novo se não existir.

d) O modo **a** abre o arquivo para escrita no final do arquivo. Não apaga o conteúdo preexistente.

e) O modo **r+** abre o arquivo para leitura e escrita. O arquivo deve existir.

1. Analise o trecho de código a seguir, escrito na linguagem C.

int x;

int y = 100;

for(x = 0; x < y;x++) {

printf("%d", x);

x = y % 2;

y -= 10;

if (y == 0) y+=10;

}

Analise as assertivas a respeito da execução do código acima.

1. Há erro(s) de sintaxe no trecho de código.
2. O único número exibido na tela (a cada execução do  
   comando printf) será o número 0.
3. A execução do laço de repetição entra em loop infinito,  
   porque x nunca terá um valor maior que y.

Está correto o que se afirma apenas em:

1. I e II
2. II e III
3. III
4. II
5. I
6. As funções de alocação dinâmica de memória são importantes  
   para a linguagem de programação C.

Desta forma, assinale a opção que contém a biblioteca  
necessária para usar as funções malloc() e free().

a) <time.h>.

b) <stdio.h>.

c) <ctype.h>.

d) <stdlib.h>.

e) <math.h>.

1. Sobre os operadores de ponteiros \*e &, analise as afirmações  
   a seguir.
2. O & é um operador unário que devolve o endereço de  
   memória do seu operando.
3. A expressão: x = \*z; faz com que a variável x seja  
   preenchida com o endereço de memória da variável z.
4. Os operadores de ponteiros & e \* têm precedência maior  
   sobre os operadores aritméticos simples de soma, subtração,  
   multiplicação e divisão.

Está correto o que se afirma apenas em:

1. I.
2. l e lI.
3. l e III.
4. II.
5. Il e IlI.
6. Bibliotecas são conjuntos de funções que foram feitas por  
   alguém e que podem ser usadas por outros programas sem  
   que nos preocupemos como código dessas funções.

Desta forma, analise o trecho de código a seguir.

#include <BIBLIOTECA>

int main() {

int r1, r2, n\_pgtos;

double a\_vista, juros, v\_pgto;

r1 = rand ();

r2 = rand ();

printf ("Números aleatórios: %d, %dnn", r1, r2);

printf (" Valor à vista “);

scanf ("%lf", &a\_vista);

printf ("Número de pagamentos: ");

scanf ("%d", &n\_pgtos);

printf (" Taxa de Juros “);

scanf ("%lf", &juros);

juros /= 100;

v\_pgto = vf (a\_vista , n\_pgtos, juros);

printf ("Valor de cada pagamento: %lfn", v\_pgto);

return 0;

}

Assinale a opção que possui o nome que pode substituir a  
palavra "biblioteca" no início do trecho do código e ela também  
possui dentre o seu conjunto de funções as scanf()e printf().

a) time.h.  
b) math.h.  
c) stdlib.h.  
d) string.h.  
e) stdio.h.

1. Analise o trecho de código em linguagem C.

int exemplo[] = {10,20,30,40};

int i,\*pont;

pont = exemplo;

for(i = 0; i< 4; i++) {

printf("%d ",\*pont++);

}

Assinale a opção que contém o comportamento correto do  
trecho de código.

1. Imprimirá os valores 10, 20, 30, 40.
2. Não imprimirá nenhum valor, pois possui um erro de sintaxe.
3. Imprimirá os valores de uma área desconhecida de memória.
4. O vetor não foi criado corretamente, pois não tem a dimensão.
5. Imprimirá os valores 11, 21, 31, 41.
6. A linguagem C oferece meios de requisitarmos espaços de  
   memória em tempo de execução. O espaço alocado  
   dinamicamente permanece reservado até que, explicitamente,  
   seja liberado pelo programa.

Desta forma, assinale a opção que possui a função  
responsável por essa liberação.

a) malloc(free());.

b) calloc().

c) maļļoc().

d) free(malloc());.

e) free().

1. Analise o código abaixo que apresenta utilização de variáveis  
   homogêneas.

# include <stdio.h >

# include <stdlib.h >

main {

int numero[10];

int i;

for (i = 0; i < 10; i+ +) {

printf ("\n %d", numero[i]);

}

system("PAUSE");  
};

Com base nas informações anteriores, assinale a opção  
correta.

a) A estrutura de repetição FOR utilizada na implementação permite que um vetor seja percorrido e seu conteúdo seja impresso.

b) Verifica-se no código implementado a utilização de uma matriz de inteiros.

c) Há um erro na linha printf( "\n%d", numero[i] ); onde utiza-se o formato (%d) incorreto.

d) Verifica-se no código implementado a utilização de uma variável chamada número que guarda o valor 10.

e) Há um erro no percurso do vetor onde a estrutura FOR deveria começar da posição 1 do vetor e não da 0 (zero).

1. As linguagens de programação são compostas por alguns tipos  
   de dados básicos.

Dessa forma, assinale a opção que não representa um tipo de  
dado básico na linguagem de programação C.

a) String.

b) char.

c) int.

d) double.

e) float.

1. Para gerar um executável em C, é preciso seguir alguns  
   passos, quando o GCC é invocado, ele normalmente realiza  
   quatro etapas para gerar o executável.

Desta forma, assinale a opção que possui a ordem correta  
desses passos.

1. Pré-Processamento, Compilação, Montagem e Linkagem (Ligação).
2. Pré-Processamento, Linkagem (Ligação), Montagem e Compilação.
3. Linkagem (Ligação), Compilação, Montagem e Pre-Processamento.
4. Pré-Processamento, Montagem, Compilação e Linkagem (Ligação).
5. Linkagem (Ligação), Montagem, Compilação e Pré-Processamento.
6. Analise o'trecho de código, escrito na linguagem C.

int a, b, m[4][4];

for (a=0; a<4; a++)

for (b=3; b>=0; b--)

m[a][b] = a+(10-pow(b,2));

Assinale a opção que contém o valor da posição m[3|2] da  
matriz, após a execução do trecho de código.

a) 9.

b) 7.

c) 8.

d) 11.

e) 10.

1. Levando em consideração a implementação de algoritmos,  
   busca-se apresentar passos a fim de resolver um problema.  
   Esses passos podem se constituir em códigos escritos em uma  
   linguagem de programação.

Analise as opções abaixo de acordo com a definição de um  
algoritmo e assinale a correta.

1. Lista de códigos escritos em uma linguagem de programação que obrigatoriamente utiliza estruturas de repetição e condição em sua implementação.
2. Sequência de comandos sempre escritos em uma linguagem de programação, orientada a objetos com o objetivo de resolver um problema.
3. Lista de códigos escritos em uma linguagem de programação que obrigatoriamente utiliza estruturas de repetição em sua implementação.
4. É uma sequência de ações infinitas, bem definidas, que resolve, em tempo finito, um problema.
5. Podemos dizer que um algoritmo não é a solução de um problema, mas o caminho até essa solução.
6. Levando em consideração as regras da linguagem C, analise  
   as asserções a seguir e a relação proposta entre elas.

I. Uma vez declarada a variável com nome conta, uma outra variável com o nome Conta não poderá ser declarada em seguida no mesmo escopo de função.

**Porque**  
II. A linguagem C é case sensitive.

A respeito dessas asserções, assinale a opção correta.

a) As asserções le ll são proposições falsas.

b) A asserção l é uma proposição verdadeira e a ll é uma proposição falsa.

c) As asserções l e ll são proposições verdadeiras e a ll é uma justificativa correta da I.

d) A asserção l é uma proposição falsa e a Il é uma proposição verdadeira.

e) As asserções l e ll são verdadeiras, mas não estabelecem relação entre si.

1. Sobre a utilização de ponteiros na linguagem C, analise as premissas abaixo:
2. Referência é quando nos referimos diretamente ao identificador do endereço da memória. A referência é dada pelo operador &.
3. Dereferência é quando nos referimos ao valor contido no endereço armazenado, ou seja, é o contrário da operação de referência. A dereferência é dada pelo operador \*.
4. Ponteiros não guardam valores, apenas endereços de memória, ou seja, se atribuímos um valor ao ponteiro, esse valor será um endereço de memória.

Está correto o que se afirma em:

1. III, apenas.
2. II, apenas.
3. I, II e III.
4. I e III, apenas.
5. II e III, apenas.
6. Elabore um programa em linguagem C que receba 100 nomes (de no máximo 100 caracteres cada) e as idades das respectivas pessoas em um vetor de estruturas (struct). Após o recebimento, listar os 100 nomes armazenados neste vetor por ordem crescente de idades.