|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **1.** | | |  | | --- | | Todo programa codificado na linguagem C pode incluir uma, ou várias bibliotecas, mas existe uma que tem que estar nessa lista, na maioria das vezes, devido a necessidade da interação com o usuário..  Assinale a alternativa que apresenta essa biblioteca. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | stdin.h |
|  |  | studio.h |
|  |  | stdio.h |
|  |  | iostream.h |
|  |  | iostream |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **2.** | | |  | | --- | | Para que possamos criar nossos programas na linguagem C, iniciamos por criar o \_\_\_\_\_\_\_\_\_\_\_\_\_\_, o qual pode ser feito através de um editor de texto não formatado e salvo com a \_\_\_\_\_\_\_\_\_\_\_\_\_\_ou mesmo utilizar um editor da linguagem, como o DEV C++. De acordo com o sistema operacional isto pode ser um pouco diferente. Após esta etapa, devemos \_\_\_\_\_\_\_\_\_\_\_ e linkeditar o programa gerando o arquivo \_\_\_\_\_\_\_\_\_\_\_\_\_, o qual após ser executado apresenta o resultado de nosso programa.  Das opções apresentadas, marque aquela que completa respectivamente e corretamente os espaços em branco destacados do texto, segundo os conceitos da linguagem C apresentados no material didático da disciplina. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Código fonte - extensão .c - compilar - executável |
|  |  | Fonte - extensão .txt - editar - executável |
|  |  | Código de texto - extensão .txt - editar - fonte |
|  |  | Executável - extensão .c - compilar - fonte |
|  |  | Código base - extensão .c - compilar - fonte |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **3.** | | |  | | --- | | Qual o nome da diretiva que é usada para incluir arquivos dentro do programa fonte que estivermos construindo? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | include |
|  |  | struct |
|  |  | typedef |
|  |  | define |
|  |  | const |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **4.** | | |  | | --- | | No ambiente Linux, não precisamos, obrigatoriamente, ter um único ambiente para editar, compilar e executar um programa codificado na linguagem C como temos no Windows. Sendo assim, qual seria a sequência para conseguirmos editar, compilar e executar um programa? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Usar um editor de texto para digitar o código/ salvar sem se preocupar com extensão/ e digitar uma linha de comando, também no terminal, usando o compilador gcc. |
|  |  | Usar um editor de texto para digitar o código e salvar com extensão c/ digitar uma linha de comando no terminal usando o compilador gcc para gerar o executável / digitar o nome do executável no terminal para executá-lo. |
|  |  | Usar um editor de texto para digitar o código/ salvar com extensão c/ digitar uma linha de comando no terminal, usando o compilador gcc que irá gerar o executável e executá-lo em seguida. |
|  |  | Usar um editor de texto para digitar o código e salvar com extensão c/ digitar uma linha de comando no terminal usando o compilador gcc para gerar o executável/ digitar o nome do executável, precedido por ./ no terminal para executá-lo. |
|  |  | Usar o terminal para digitar o código/ salvar sem se preocupar com extensão/ e digitar uma linha de comando, também no terminal, usando o compilador gcc. |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **5.** | | |  | | --- | | Qual das características abaixo **NÃO** pode ser atribuída á linguagem C? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Modularidade |
|  |  | Rapidez, pois gera executável compacto |
|  |  | Estruturada |
|  |  | Complexidade porque tem muitas palavras reservadas |
|  |  | Portabilidade |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **6.** | | |  | | --- | | Quando precisamos compilar um código na linguagem C na plataforma Linux ,caso não tenhamos um ambiente onde se possa editar e compilar, podemos fazer esse processo no terminal. Um desenvolvedor experimentou as linhas de comando abaixo. 1) gcc -o nomeExecutável nomedoFonte.c  2) gcc -o nomedoFonte.c Analise as opções e escolha a correta. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Não será criado um executável na segunda forma, pois aparecerá uma mensagem de erro informando falta de parâmetros. |
|  |  | Ambas linhas de comando não geram executáveis. |
|  |  | Será criado um executável com um nome fixo independente do nome do fonte na segunda forma. |
|  |  | Sempre um executável com o mesmo nome do fonte será criado em ambos os casos. |
|  |  | O compilador gcc não serve para linguagem C. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **7.** | | |  | | --- | | Observe o programa abaixo e selecione a resposta que mostra e/ou explica o que será exibido no display.  #include  int main() { printf ("O valor eh = \\n%04d ", 12); } | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | O valor eh = em uma linha e na linha abaixo 12 |
|  |  | O valor eh = em uma linha e na linha abaixo 0012 |
|  |  | O valor eh = \n0012 |
|  |  | O valor eh = \n 12 |
|  |  | O valor eh = 000012 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **8.** | | |  | | --- | | Um Engenheiro estava construindo um trecho de programa onde pedia que o usuário respondesse com uma letra ou um algarismo. Se o usuário não tivesse respondido, ele enviaria uma mensagem de erro. Ele sabia que precisava aprender mais alguma estrutura, mas, no momento, pensou em experimentar uma função que pudesse lhe dar um retorno. Pesquisou nas bibliotecas estudadas na aula 2 e achou. Qual o nome dessa função? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | isalpha() |
|  |  | isspace() |
|  |  | isalnum() |
|  |  | isdigit() |
|  |  | isnotalphadigit() |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **9.** | | |  | | --- | | Observe o programa abaixo e selecione a resposta que mostra e/ou explica o que será exibido no display.  #include  int main()  {  printf ("%s%d%%","Juros de ",10); } | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | %s%d%% Juros de 10 |
|  |  | % Juros de 10 |
|  |  | Nada porque tem muitos % |
|  |  | Juros de 10% |
|  |  | 10 Juros de |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **10.** | | |  | | --- | | Um Engenheiro Civil precisava usar uma função que pudesse arredondar para cima, uma vez que ao compramos pisos/ azulejos sempre usamos um valor inteiro. Observe as funções abaixo e escolha a função capaz de resolver esse problema. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | rounder() |
|  |  | floor() |
|  |  | pow() |
|  |  | sqrt() |
|  |  | ceil() |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **11.** | | |  | | --- | | Um Engenheiro Mecânico tinha um arquivo onde ele gostaria que os nomes dos materiais fossem todos convertidos para letras minúsculas. Observe as linhas abaixo e assinale a opção que contem a função que realiza essa conversão e a biblioteca a qual pertence. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | toupper() e stdlib.h |
|  |  | tolower() e ctype.h |
|  |  | atof() e stdlib.h |
|  |  | atoi() e math.h |
|  |  | isalpha() e stdlib.h |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **12.** | | |  | | --- | | Um Engenheiro Ambiental tinha um arquivo onde ele gostaria retirar algarismos que estavam dentro de várias cadeias de caracteres e que fossem todos convertidos para números reais, uma vez que eles representavam tamanhos de árvores. Observe as linhas abaixo e assinale a opção que contem a função que realiza essa conversão e a biblioteca a qual pertence. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | atoi() e math.h |
|  |  | isalpha() e stdlib.h |
|  |  | toupper() e stdlib.h |
|  |  | tolower() e ctype.h |
|  |  | atof() e stdlib.h |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **13.** | | |  | | --- | | Quando precisamos incrementar o valor de uma variável em C, geralmente usamos a forma contrata, com os operadores pré ou pós fixados de acordo com a necessidade.  Supondo as seguintes definições e desconsiderando a necessidade do restante do programa:  Int a = 5; Int b = 8;  Após a execução das operações na ordem em que são apresentadas, Marque a resposta **correta**: | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | a += b++; o valor de a e b são respectivamente: 9 e 13 |
|  |  | a += b++; o valor de a e b são respectivamente: 14 e 9 |
|  |  | a += ++b; o valor de a e b são respectivamente: 14 e 8 |
|  |  | a = ++b; o valor de a e b são respectivamente: 8 e 9 |
|  |  | a = b++; o valor de a e b são respectivamente: 8 e 9 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **14.** | | |  | | --- | | Observe abaixo várias funções que recebem dados via teclado. Qual delas permite uma leitura formatada? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | fgets() |
|  |  | scanf() |
|  |  | gets() |
|  |  | getchar() |
|  |  | fgetc() |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **15.** | | |  | | --- | | Na programação em C como em outras linguagens, muitas vezes se faz necessário a conversão de tipos de dados, para que possamos exibir ou armazenar o resultado de uma operação da forma desejada. O nome dado a esta operação é Cast.  Das opções apresentadas, marque aquela que converte e armazena corretamente o valor de uma variável **n** do tipo **int** para a variável **x** do tipo **float** utilizando o Cast. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | x = (int) n; |
|  |  | x = (float) n; |
|  |  | n = float x; |
|  |  | n = (float) x; |
|  |  | x = float n; |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **16.** | | |  | | --- | | Na linguagem de programação C, como em outras, na criação de programas, devemos definir os tipos de dados para nossas variáveis de acordo com o problema em questão.  Das opções apresentadas qual possui apenas tipos da linguagem C, conforme o material apresentado nas aulas. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | char, int, real, double, void |
|  |  | char, int, float, double, void |
|  |  | char, integer, float, real, boolean |
|  |  | char, int, float, double, boolean |
|  |  | char, int, real, double, boolean |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **17.** | | |  | | --- | | Observe abaixo algumas declarações de variáveis e assinale a **declaração errada**. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | long matricula; |
|  |  | long double profundidade; |
|  |  | long float x; |
|  |  | int i; |
|  |  | short int a; |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **18.** | | |  | | --- | | Escolha a opção que inclui somente nomes válidos para variáveis na linguagem C. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | nota-2 , a36, x\_y |
|  |  | 2\_ou\_1, fim, \*h, j |
|  |  | If, nota\_do\_aluno\_2, HOJE, \_yes |
|  |  | 2nota, a36, x#y, --j |
|  |  | i, j, int, obs |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **19.** | | |  | | --- | | Na construção de programas na linguagem C, precisamos tomar muito cuidado com a sintaxe das expressões condicionais, para que não tenhamos resultados inesperados.  Supondo que as variáveis foram devidamente declaradas e levando em conta apenas a linha de código apresentada, dentre as opções qual **apresenta erro** na sintaxe para a linguagem C. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | if (a > 0) {a = 2;} else {b= 2;} |
|  |  | if (a > 0) {a = 2;} else if (b < 10) {b= 2;} |
|  |  | if (x > 0) a = 2 else b = 2 ; |
|  |  | if x > 0 { a = 2;} |
|  |  | if (a > 0) { a = 2;} |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **20.** | | |  | | --- | | Um desenvolvedor estava construindo um código que tinha quatro possibilidades de resposta válidas além de uma mensagem de erro. O dado testado era do tipo inteiro. Assinale a opção que ele escolheu. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | quatro ifs simples |
|  |  | quatro ifs simples e um composto |
|  |  | switch() com quatro cases e um default |
|  |  | três ifs simples e um composto |
|  |  | switch() com cinco cases |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **21.** | | |  | | --- | | Existe sempre uma dúvida por parte do desenvolvedor se escolhe switch() ou if quando está usando a Linguagem C. Sabemos que existem algumas exigências para se usar o switch(). Analise as opções abaixo e assinale a opção correta.  I Qualquer tipo pode ser usado.  II Somente o tipo vetor de char que Não pode ser usado.  III Não pode ser usado intervalo tipo: 1..3 em um case.  IV Sempre será obrigatório usar o comando break; ao final de um case.  V Quando o valor digitado não estiver presente em nenhum dos cases, irá para a opção default, caso exista. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | II, IV e V são verdadeiras |
|  |  | I, III e IV são verdadeiras |
|  |  | III e V são verdadeiras |
|  |  | I, III, IV e V são verdadeiras |
|  |  | II, III e IV são verdadeiras |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
|  |  |  |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | I, II e III estão corretas |
|  |  | II e IV estão corretas |
|  |  | II, III e IV estão corretas |
|  |  | I, II, IV e V estão corretas |
|  |  | I, II e V estão corretas |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **22.** | | |  | | --- | | O desenvolvimento de um programa precisa atender as necessidades do usuário final, para isto o desenvolvedor conta com algumas estruturas que facilitam a construção de seu código fonte. Uma delas é a estrutura condicional **if**.  Das opções apresentadas marque a qual **melhor define** esta estrutura. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Compara os valores possíveis para uma variável e dependendo do resultado permite que o programa siga por caminhos diferentes. |
|  |  | Testa uma condição e dependendo do resultado permite que o programa siga por caminhos diferentes repetidamente. |
|  |  | Compara os valores de algumas variáveis e dependendo do resultado permite que o programa siga por caminhos diferentes. |
|  |  | Compara os valores de algumas variáveis dependendo do resultado permite que o programa possa fazer um caminho diferente repetidamente. |
|  |  | Testa uma expressão e dependendo do resultado permite que o programa siga por caminhos diferentes. |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **23.** | | |  | | --- | | Preencha as lacunas abaixo para que a mensagem abaixo seja exibida, caso a variável Idade seja maior ou igual a 18 e a variável Peso seja maior que 50.   if((Idade\_ \_ \_18)\_ \_ \_(peso\_ \_ \_50))  { printf("Você pode ser doador de Sangue"); } | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | >=; &; >; |
|  |  | >>; and; >; |
|  |  | <>; and; >; |
|  |  | >=; e; >; |
|  |  | >=; &&; >; |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | |  | | --- | |  | |
|  | |

|  |
| --- |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **24.** | | |  | | --- | | Para o desenvolvimento de nossos programa na linguagem C, contamos com 3 tipos básicos de estruturas de repetição, for, while e do..while. O funcionamento e aplicação dos mesmos tem pequenas diferenças.      int i;      for(i=0; i < 10; i+=2)          if (i%2)          printf("%d",i);     printf("%d",i);  Supondo que as demais  linhas necessárias para o funcionamento do programa estejam devidamente criadas e baseado no trecho de código apresentado, o que será **exatamente apresentado na tela**após a sua execução. | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | 02468 |
|  |  | 0 2 4 6 8 10 |
|  |  | 0246810 |
|  |  | 10 |
|  |  | 024681010 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **25.** | | |  | | --- | | Para o programa abaixo qual a saída apresentada?  main(){ int x,soma=0; for(x=10; x<=15;x++) {   printf(" %d",pow(x,2)); } system("PAUSE"); return 0; } | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | 9 10 11 12 13 14 15 |
|  |  | 20 22 24 26 28 30 |
|  |  | 100 121 144 169 196 225 |
|  |  | 10 11 12 13 14 15 |
|  |  | 15 14 13 12 11 10 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **26.** | | |  | | --- | | Uma das principais características que consolidaram o sucesso na utilização dos computadores para a resolução de problemas foi a sua capacidade de repetir o processamento de um conjunto de operações para grandes quantidades de dados. Exemplos de conjuntos de tarefas que repetimos diversas vezes dentro de uma situação específica podem ser observados largamente no nosso dia a dia. As estruturas de repetição proveem uma maneira de repetir um conjunto de procedimentos até que determinado objetivo seja atingido, quando a repetição se encerra. Todas as estruturas de repetição têm em comum o fato de haver uma condição de controle, expressa através de uma expressão lógica, que é testada em cada ciclo para determinar se a repetição prossegue ou não. Tendo como base o código abaixo, qual opção que demonstra o que será exibido na tela?   void main() { int i = 2; int j = 6; for(i=0;i<=5;i++){ do{  printf("%d", ++i); }while(++i <= j--); } system("PAUSE"); } | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | 23456 |
|  |  | 256 |
|  |  | 246 |
|  |  | 135 |
|  |  | 265 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **27.** | | |  | | --- | | Qual a estrutura de repetição usada para quantidade indeterminada de repetições? | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | while |
|  |  | if-else |
|  |  | for |
|  |  | switch |
|  |  | struct |

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
| |  | | --- | | **28.** | | |  | | --- | | Durante a criação de um programa, foi percebido que uma parte do código era executada várias vezes, porém não era em uma sequência de repetições e sim esporadicamente. Cada vez que esta parte era executada inicialmente precisava-se de um valor diferente e no final do trecho a resposta nem sempre era a mesma. Era necessário achar alguma forma de evitar a redundância de códigos e melhorar a escrita e desempenho do sistema. Baseado na descrição apresentada acima e utilizando-se da melhor forma os conceitos de programação, qual seria a melhor solução pra o problema: | | |  | | --- | |  | |
|  | | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Utilizar a estrutura de loop FOR com o GOTO, para executar a parte do código que se repetia. |
|  |  | Utilizar a estrutura de loop FOR, para executar a parte do código que se repetia. |
|  |  | Criar uma função para executar a parte do código que se repetia. |
|  |  | Utilizar a estrutura de loop DO-WHILE, para executar a parte do código que se repetia. |
|  |  | Utilizar a estrutura de loop WHILE, para executar a parte do código que se repetia. |