Revisão da linguagem C

01. A função **main()** deve existir em alguma parte de um programa em C e marca o ponto de início da execução.

A.	Verdadeiro
B.	Falso

02. Em um programa em C, os nomes **num** e **Num** podem ser usados indistintamente em diferentes partes do programa para referenciar a mesma variável.

A.	Verdadeiro
B.	Falso

03.A opção que inclui apenas nomes válidos para variáveis na linguagem C é:

A.	If, a_b_2, H789, _yes
B.	i, j, int, obs
C.	9xy, a36, x*y, --j
D.	2_ou_1, \fim, *h, j
E.	Nenhuma das opções anteriores

Sobre o trecho de programa abaixo

```
… int main()
      {
          char opcao;
          int i = 1;
    opcao = 'B';
    ...
 }
```

pode-se afirmar que é:

A.	Válido na linguagem C
B.	Não válido na linguagem C

04. Em C, **"v"** e **'v'** representam a mesma constante.

A.	Verdadeiro
B.	Falso

05. O programa

```
#include <stdio.h>
main()
     {
        int numero;
        scanf("%d",&numero);
        printf("%d",numero);
     }
```

Lê uma variável pelo teclado e a imprime na tela.

A.	Verdadeiro
B.	Falso

06. A instrução ***#include <stdio.h>*** no programa anterior é colocada para que se possa utilizar funções tais como **scanf** e **printf** .

A.	Verdadeiro
B.	Falso

07. Na linguagem C, cada comentário deve ser restrito a uma única linha de código.

A.	Verdadeiro
B.	Falso

08. O programa a seguir está correto.

```
int main()
{
      int x=3; y=5, z=7;
      printf("Os números são: %d %d %d\n,x,y,z,w)
}
```

A.	Verdadeiro
B.	Falso (qual o erro?) R:

09. Textos delimitados por ******* (início) e ******* (término) são ignorados pelo computador na linguagem C.

A.	Verdadeiro
B.	Falso

10. O que faz o seguinte programa em C?

```
#include <stdio.h>
main()
{
   int vlr =6;
   printf ("\n Valor = %d ", vlr);
}
```

A.	Nada
B.	Imprime: Valor = 6
C.	Imprime: \n O valor de vlr = %d
D.	Pula para a próxima linha e imprime: Valor = 6
E.	Nenhuma das alternativas anteriores está correta.

11. Programas codificados em C devem conter pelo menos a função *main()*.

A.	Verdadeiro
B.	Falso

12. Em C, pares de chaves (***{}***) servem sempre de delimitadores para blocos de código.

A.	Verdadeiro
B.	Falso

13. Em C, uma linha inteira de código equivale a um **comando**, devendo ser encerrada com um ponto-e-vírgula (;).

A.	Verdadeiro
B.	Falso

14. Qual a saída produzida pelo trecho de código a seguir:

```
int x;
for (x = 35 ; x > 0 ; x=x/3)
    printf("%d " , x) ;
```

A.	35 11 3 1
B.	11 3 1
C.	11 3 1 0
D.	35 11 3
E.	Nenhuma das opções anteriores

15. Caso o nome da função seja escrito incorretamente em um programa em C, o *linker* indicará para o programador o erro de digitação e lista o conteúdo da biblioteca na qual a função se encontra, a fim de que o programador digite corretamente o nome da função.

A.	Verdadeiro
B.	Falso

16. O trecho de código abaixo

```
#include stdio.h
int main()
{
    int i1;
        printf("Entre com o primeiro valor:");
        scanf( "%d", &i1 );
        printf( "O valor digitado foi %d\n", i1 );
    return 0;
}
```

A.	Imprimirá na tela uma mensagem para a entrada de um valor e receberá o valor do teclado, imprimindo-o na tela sem mais nada.
B.	Imprimirá na tela uma mensagem para a entrada de um valor e receberá o valor do teclado, imprimindo a mensagem "O valor digitado foi" seguido do valor digitado, por sua vez seguido do símbolo %.
C.	Imprimirá na tela uma mensagem para a entrada de um valor e receberá o valor do teclado, imprimindo a mensagem "O valor digitado foi" seguido do valor digitado.
D.	Imprimirá na tela uma mensagem para a entrada de um valor e, em seguida, será encerrado.
E.	Nenhuma das opções anteriores

17. Se um comando executável referenciar uma variável que não foi anteriormente declarada, será produzido um erro de sintaxe:

A.	Verdadeiro
B.	Falso

18. O seguinte trecho de código

```
int x,y;
int a = 14, b = 3;
x = a/b;
y = a%b;
z = x/y;
```

gerará como resultados:

A.	x = 4.66666, y = 2 e z = 2
B.	x = 4, y = 0.66666 e z = 2
C.	x = 4, y = 2 e z = 2
D.	x = 4.66666, y = 0.66666 e z = 2
E.	Nenhuma das alternativas anteriores

19. Seja o seguinte trecho de programa:

```
int i=3,j=5;
int *p, *q;
p = &i;
q = &j;
```

Qual é o valor das seguintes expressões ?

a) `p == &i;` b) `*p - *q;` c) `**&p` d) `3* - *p/(*q)+7`

20. Qual é o resultado do seguinte programa? Suponha o endereço de memória inicial de vet 4000

```
#include <conio.h>
#include <stdio.h>
int main(){
    float vet[5] = {1.1,2.2,3.3,4.4,5.5};
    float *f;
    int i;
    f = vet;
    printf("contador/valor/valor/endereco/endereco");
    for(i = 0 ; i <= 4 ; i++){
        printf("\ni = %d",i);
        printf("   vet[%d] = %.1f",i, vet[i]);
        printf("   *(f + %d) = %.1f",i, *(f+i));
        printf("   &vet[%d] = %X",i, &vet[i]);
        printf("   (f + %d) = %X",i, f+i);
    }
return 0; }
```

21. Assumindo que **pulo[]** é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento da matriz?

a) *(pulo + 2) b) *(pulo + 4) c) pulo + 4 d) pulo + 2

22. O que fazem os seguintes programas?

```
#include <conio.h>
#include <stdio.h>
int main(){
  int vet[] = {4,9,13};
  int i;
  for(i=0;i<3;i++){
    printf("%d ",*(vet+i));
  }
return 0;
}
```

```
#include <conio.h>
#include <stdio.h>
int main(){
  int vet[] = {4,9,13};
  int i;
  for(i=0;i<3;i++){
    printf("%p ",vet+i);
  }
return 0;
}
```

23. Seja **vet** um vetor de 4 elementos: **TIPO vet[4]**. Supor que depois da declaração, **vet** esteja armazenado no endereço de memória `4092` (ou seja, o endereço de `vet[0]`). Supor também que na máquina usada uma variável do tipo `char` ocupa 1 byte, do tipo `int` ocupa 2 bytes, do tipo `float` ocupa 4 bytes e do tipo `double` ocupa 8 bytes.

Qual o valor de `vet+1`, `vet+2` e `vet+3` se:

a) **vet** for declarado como `char`?

b) **vet** for declarado como `int`?

c) **vet** for declarado como `float`?

d) **vet** for declarado como `double`?

24. Faça uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume ($v = 4/3.P .R^3$)

25. Faça uma função que recebe por parâmetro um valor inteiro e positivo e retorna o valor 'V' caso o valor seja primo e 'F' em caso contrário.