

Aluno(a): .....

1. Preencha as lacunas do programa abaixo para que, após sua execução, o valor de cada posição do vetor v corresponda ao triplo do índice daquela posição

```
1 int main(){
2     int v[5];
3     int i;
4     for(_____;i++){
5         _____ ;
6     }
7 }
```

Linha 4: \_\_\_\_\_

Linha 5: \_\_\_\_\_

2. Preencha as lacunas do programa abaixo para que, após sua execução, o vetor v2 contenha os mesmos elementos do vetor v1, porém em ordem inversa.

```
1 int main(){
2     int v1[5] = {1,2,3,4,5};
3     int v2[5];
4     int i;
5     for(_____;i++){
6         v2[_____] = v1[_____];
7     }
8 }
```

Linha 5: \_\_\_\_\_

Linha 6 (1): \_\_\_\_\_

Linha 6 (2): \_\_\_\_\_

3. Preencha as lacunas do programa abaixo para que, ao final, seja impresso:

```
4 6
3 2
7 9
```

```
1 int main(){
2     int v[3][2] = {{4,6},{3,2},{7,9}};
3     int x, y;
4     for(_____;x++){
5         for(_____;y++){
6             printf("%d", _____ );
7         }
8         printf("\n");
9     }
10 }
```

Linha 4: \_\_\_\_\_

Linha 5: \_\_\_\_\_

Linha 6: \_\_\_\_\_

4. Preencha as lacunas do programa abaixo para que, ao final, seja impresso

```
8 2 7
```

```
1 #include<stdio.h>
2
3 int main(){
4     int v[3][3]={{4,6,8},{3,2,1},{7,9,5}};
5     int x;
6     for (_____;x++){
7         printf("%d",_____);
8     }
9     printf ("\n");
10 }
```

Linha 6: \_\_\_\_\_

Linha 7: \_\_\_\_\_

5. Qual será o valor da variável y ao final da execução do programa abaixo?

```
void a(int p){
    int x = p*2;
}

int b(int p){
    return p+c(p);
}

int c(int p){
    return p*p;
}

int main(){
    int x = 5, y;
    a(x);
    y = x + b(x);
}
```

Resposta: \_\_\_\_\_

6. Qual será o valor da variável w ao final da execução do programa abaixo?

```
int a(int i, int j){
    int k = i;
    i = j;
    j = k;
    return i*3-j*2;
}

int main(){
    int x = 2, y=3, z, w;
    z = a(x,y);
    w = x + y + z;
}
```

Resposta: \_\_\_\_\_

7. O que será impresso pelo programa abaixo?

```
#include <stdio.h>
#include <string.h>

int main () {
    char s[80] = "ufsc";
    strcat(s,s);
    int tamanho = strlen(s);
    for (int i = 0; i <= tamanho; i=i+2){
        printf ( "%c" ,s[tamanho-i]);
    }
}
```

Resposta: \_\_\_\_\_

8. Qual será o valor da variável i ao final da execução do programa a seguir?

```
#include <stdio.h>
#include <string.h>

int main () {
    char s[80] = "engenharia";
    char w[80] = "ufsc";
    int i=0;
    while(strcmp(s,w)!=0){
        w[i] = s[i];
        i++;
    }
}
```

Resposta: \_\_\_\_\_

9. Considere o programa abaixo e as afirmações a seguir:

```
int main(){
    FILE *arquivo;
    arquivo = fopen("ufsc.txt","w");
    fputs("primeira \n linha",arquivo);
    fclose(arquivo);
}
```

- (I) se o arquivo “ufsc.txt” não existir, o programa retornará uma mensagem de erro
  - (II) o programa introduzirá uma única linha no arquivo “ufsc.txt”
  - (III) o parâmetro “w” pode ser substituído por “r” sem provocar erros na execução do programa
- (a) Todas as afirmações estão corretas
  - (b) Todas as afirmações estão erradas
  - (c) Apenas as afirmações (I) e (II) estão corretas
  - (d) Apenas as afirmações (I) e (III) estão corretas
  - (e) Apenas a afirmação (III) está correta

10. A modularização de algoritmos é importante para organizar melhor o código e facilitar a manutenção, entre outras coisas. Sobre funções e procedimentos, assinale a alternativa CORRETA sobre a modularização:

- (a) O procedimento sempre retorna um valor ao programa.
- (b) A função retorna um valor ao programa.
- (c) As variáveis definidas no escopo de cada função são acessíveis em todo o programa.
- (d) As variáveis locais são declaradas no escopo do programa inteiro.