

Aluno(a): .....

1. O que será impresso pelo programa abaixo?

```
1 int main() {
2     int x=2, y=3, z, *t, *w;
3     t = &x;
4     w = &y;
5     z = *t;
6     *t = *w;
7     *w = z;
8     printf("%d,%d",x,y);
9 }
```

Resposta: .....

2. Qual o valor da variável  $j$  ao final da execução do programa abaixo?

```
1 int main(){
2     int v[5] = {1,2,3,4,5};
3     int j = 0;
4     for(int i=0;i<5;i++){
5         j = j + *(v+i) + *(v+4-i);
6     }
7 }
```

Resposta: .....

3. No programa a seguir, a variável  $p$  deve apontar para um bloco de memória que armazena os 10 primeiros múltiplos de 3 (isto é, o resultado de  $3 \times 1, \dots, 3 \times 10$ ). Preencha as lacunas para que o programa preencha esse bloco de memória adequadamente e imprima o conteúdo armazenado nele.

```
1 int main(){
2     int i,*p;
3     p = _____;
4     for(i=0;i<10;i++){
5         _____;
6     }
7     for(i=0;i<10;i++){
8         printf("%d", _____);
9     }
10 }
```

Linha 3: .....

Linha 5: .....

Linha 8: .....

4. O que será impresso pelo programa abaixo?

```
1 int main(){
2     int i, *w;
3     w = malloc(sizeof(int)*3);
4     for(i=0;i<3;i++){
5         *(w+i) = i+1;
6     }
7     for(i=0;i<3;i++){
8         printf("%d ", *(w+i)+1);
9     }
10 }
```

Resposta: .....

5. Qual das seguintes expressões referenciam o valor do terceiro elemento (elemento de índice 2) do vetor  $v$ ?

- (a)  $v++$ ;      (d)  $(v + 4)$       (g)  $\&(v + 2)$   
(b)  $*(v + 2)$       (e)  $(v + 2)$   
(c)  $*(v + 4)$       (f)  $\&(v + 4)$

6. Qual será o retorno da função  $f(n)$  para  $n=3$ ?

```
1 int f(n){
2     if(n==0)
3         return 1;
4     else
5         return f(n-1)+f(n-1);
6 }
```

Resposta: .....

7. Qual será o valor da variável  $x$  após a execução do programa abaixo? ?

```
1 int f(int *v, int s){
2     if(s==1)
3         return (*v)*(*v);
4     else
5         return (*v)*(*v)+f(v+1,s-1);
6 }
7
8 int main(){
9     int v[4] = {1,2,3,4};
10    int x = f(v,4);
11    printf("%d\n",x);
```

Resposta: .....

8. Considere as descrições enumeradas a seguir e numere os itens a seguir.

- 1 Estrutura de dados que pode ser percorrida em ambos os sentidos pois cada um de seus elementos aponta tanto para seu predecessor quanto para o sucessor.
- 2 Pode-se acessar o primeiro elemento da lista diretamente a partir do último elemento, sem percorrer toda a lista.
- 3 A partir de um determinado elemento da lista, só é possível acessar os seus sucessores pois os elementos não possuem apontamento para seus predecessores.
- 4 Requer que os elementos sejam movidos para posições anteriores ou posteriores no caso de exclusões e inclusões de elementos da lista;

- ( ) Listas baseadas em vetores dinâmicos  
( ) Listas encadeadas  
( ) Listas duplamente encadeadas  
( ) Listas circulares

Para as questões 9 a 10, considere a seguinte estrutura:

```
    p = p->next;
}
if(p!=NULL){
    if(anterior==NULL)
        lista = _____;
```

9. Complete as lacunas da função abaixo para que ela faça a inserção de um item no *fim* da lista

```
1 listaItem *insere_fim(listaItem *lista,int dado){
2     listaItem *novo = malloc(_____);
3     novo->dado = dado;
4     novo->next = NULL;
5     if(lista==NULL){
6         _____;
7     }
8     else{
9         listaItem *ultimo = lista;
10        while(_____){
11            ultimo = _____;
12        }
13        _____ = novo;
14    }
15    return lista;
16 }
```

Linha 2: \_\_\_\_\_  
Linha 6: \_\_\_\_\_  
Linha 10: \_\_\_\_\_  
Linha 11: \_\_\_\_\_  
Linha 13: \_\_\_\_\_

10. Complete as lacunas da função abaixo para que ela faça a inserção de um item no *início* da lista

```
1 listaItem *insere_inicio(listaItem *lista,int dado){
2     listaItem *novo = _____;
3     novo->dado = dado;
4     novo->next = _____;
5     _____;
6     return lista;
7 }
```

Linha 2: \_\_\_\_\_  
Linha 4: \_\_\_\_\_  
Linha 5: \_\_\_\_\_