

Aluno(a):

1. O que será impresso pelo programa abaixo?

```
1 #include <stdio.h>
2
3 int main() {
4     int x, *y, z;
5     x = 2;
6     y = &x;
7     (*y)++;
8     z = (x++)*2;
9     printf("%d,%d,%d",x,*y,z);
10 }
```

Resposta:

2. O que será impresso pelo programa abaixo?

```
1 #include <stdio.h>
2
3 int main(){
4     int v[5] = {3,5,4,7,1};
5     for(int i=1;i<5;i++){
6         printf("%d-", *(v+i)**(v+i-1));
7     }
8 }
```

Resposta:

3. O que será impresso pelo programa abaixo?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int i,*p;
6     p = malloc(sizeof(int)*3);
7     for(i=0;i<3;i++){
8         *(p+i) = i+1;
9     }
10    for(i=0;i<3;i++){
11        printf("%d", *(p+i)+1);
12    }
13 }
```

Resposta:

4. O programa abaixo deve armazenar inteiros em um vetor v cujo tamanho é informado pelo usuário na linha 7. Informe como devem ser preenchidas as lacunas nas linhas 8, 11 e 14 para que o vetor seja criado, para que os números informados pelo usuário sejam armazenados no vetor e para que os valores lidos sejam impressos na ordem em que foram lidos.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int *v, q, i;
6     printf("Informe o tamanho do vetor: ");
7     scanf("%d",&q);
8     v = malloc(sizeof(int)*q);
9     for(int i=0;i<q;i++){
10         printf("Informe o %do valor: ",i+1);
11         scanf("%d",&v[i]);
12     }
13     for(i=0;i<q;i++){
14         printf("%d\n", v[i]);
15     }
16 }
```

Respostas:

- Linha 8:
— Linha 11:
— Linha 14:

5. Qual será o retorno da função f abaixo se ela receber parâmetros x=4 e y=5?

```
1 int f(int x, int y){
2     if(y==2)
3         return x * x;
4     else
5         return x * f(x,y-1);
6 }
```

Resposta:

6. O que será impresso pelo programa abaixo?

```
1 #include<stdio.h>
2
3 int f(int *v, int s){
4     if(s==1)
5         return *v;
6     else
7         return *v + f(v+1,s-1);
8 }
9
10 int main(){
11     int v[5] = {3,6,2,6,4};
12     printf("%d",f(v,5));
13 }
```

Resposta:

Para as questões 7 e 8, considere a seguinte estrutura:

```
typedef struct lstItem{
    int dado;
    struct lstItem *next;
} listaItem;
```

7. Complete as lacunas das linhas 4 e 5 de modo que a função abaixo faça a inserção de um elemento no início da lista:

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

Linha 4:

Linha 5:

8. Complete as lacunas das linhas 10 e 13 de modo que a função abaixo faça a inserção de um elemento no fim da lista. Assuma que a lista *não* está vazia.

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

Linha 10:

Linha 13:

9. 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
10. Qual das seguintes expressões referenciam o valor do terceiro elemento (elemento de índice 2) do vetor v ?
- (a) $*(v + 2)$ (c) $v + 4$ (e) $v++$;
(b) $*(v + 4)$ (d) $v + 2$