Aluno(a): .....

1. O que será impresso pelo programa abaixo?

```
#include <stdio.h>
int main() {
   int x, *y, z;
   x = 2;
   y = &x;
   (*y)++;
   z = (x++)*2;
   printf("%d,%d,%d",x,*y,z);
}
Resposta:
```

2. No programa abaixo, o usuário informa o tamanho de um vetor a ser criado. Esse tamanho é armazenado na variável  ${\bf q}$ . O vetor deve armazenar os "q" múltplos de 5  $(5\times 1, 5\times 2, \cdots, 5\times q)$ . Em seguida o programa deve imprimir os elementos do vetor. Preencha as lacunas abaixo para que o programa tenha esse comportamento.

```
1 int main(){
     int *v, q, i;
printf("Tamanho do vetor: ");
3
     scanf("%d",&q);
4
     v = _____
     for(int i=0;i<q;i++){</pre>
6
7
        = (q+1)*5;
8
     for(i=0;i<q;i++){
9
         printf("%d\n", ____);
10
11
12 }
```

Linha 5:	
Linha 7:	
Linha 10:	

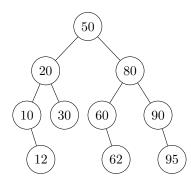
3. Complete as lacunas da função a seguir para que ela faça a inserção de um elemento no fim da lista. Assuma que a lista  $n\tilde{ao}$  está vazia.

```
1 typedef struct lstItem{
     int dado;
     struct lstItem *next;
4 }listaItem;
6 listaItem *ins_fim(listaItem *lista,
                         int dado){
8
     listaItem *novo = ____;
     novo->dado = dado;
9
    novo->next = NULL;
10
     if(lista==NULL){
11
       lista = novo;
12
13
15
        listaItem *ultimo = lista;
16
        while(_____){
17
         ultimo = ultimo->next;
18
19
        -----;
20
     return lista:
21
```

Linha 8: \_\_\_\_\_\_ Linha 16: \_\_\_\_\_ Linha 19: \_\_\_\_\_

- 4. Considere a seguinte sequência de operações: push(9), push(3), pop(), push(2), pop(), push(4), push(7), pop(), pop(), push(6), push(1), pop(), push(2), push(5), pop().
  - A soma dos elementos remanescentes em uma *fila*, inicialmente vazia, após essa sequência de operações será \_\_\_\_\_.
  - A soma dos elementos remanescentes em uma *pilha*, inicialmente vazia, após essa sequência de operações será \_\_\_\_\_\_.
- 5. Considere uma tabela hash com 23 espaços indexados de 0 a 22, utilizando a função hash  $h(x) = x \mod 23$  (método da divisão). Nesta tabela, são inseridas, na ordem dada, as seguntes chaves: 44, 46, 49, 70, 27, 71, 90, 97, 95. Esboce a tabela hash após a inserção das chaves, considerando solução de colisões por encadeamento. Utilize o formato [indice]:{chaves}.

6. Considere a árvore a seguir:



Escreva abaixo a sequência de números que será impressa caso a árvore seja percorrida

- a) Em pré-ordem: \_\_\_\_\_
- b) Em ordem: \_\_\_\_\_
- c) Em pós-ordem: \_\_\_\_\_
- 7. Desenhe a árvore binária de pesquisa gerada pela inserção, em sequência, dos seguintes números: 20, 30, 10, 40, 5, 21, 15, 25, 35, 12, 7, 23.

8.	Considere o vetor $v = [2, 6, 18, 19, 24, 27, 37, 43, 56, 99]$ . Escreva a sequência de elementos avaliados na busca binária pelos seguintes números:  • 19:	10.	Aplique o algoritmo de ordenação $Quick\ sort$ no vetor $v=[30,68,36,49,61,58,94,78]$ utilizando como pivô $p$ o elemento central do vetor (ou seja, sendo $l$ e $r$ os índices das extremidades esquerda e direita do vetor
9.	• 37: • 47: Escreva o estado do vetor $v=[74,91,5,3,9,43,55,71,58,20] \text{ após cada um dos passos da execução do algoritmo de ordenação por seleção.}$	ordenação, o ser ordenado cada um dos o vetor orden ordenação est	respectivamente, considera-se $p = \lfloor \frac{l+r}{2} \rfloor$ ). Durante a ordenação, o subvetor mais à esquerda do pivô deve ser ordenado antes do subvetor mais à direita. Mostre cada um dos passos da ordenação que levaram a obter o vetor ordenado. Considere que um passo de ordenação está completo quando o pivô está em sua posição definitiva.

## Informações úteis

- Em vetores com número par de elementos, considerar, como elemento central, o último elemento da primeira metade.
- Quando dois subvetores precisarem ser ordenados, considerar que o subvetor da esquerda é ordenado antes do subvetor da direita.
- Ao dividir um vetor  $v = [v_0, \cdots, v_n]$  pela metade, sendo 0 (zero) o índice do primeiro elemento e n o índice do último elemento, considerar que (i) a primeira metade é  $[v_0, \cdots, v_c]$  e (ii) a segunda metade é  $[v_{c+1}, \cdots, v_n]$ , onde  $c = \lfloor \frac{0+n}{2} \rfloor$ .
- Em um algoritmo de ordenação, um passo completo acontece quando um determinado número do vetor é colocado em sua posição apropriada. Por exmplo, considerando o número 90 no vetor v = [90, 50, 30], ao final de um passo completo, o vetor teria a seguinte ordem: v = [50, 30, 90].