Universidade Federal de Santa Catarina - Campus Blumenau
Departamento de Engenharia de Controle, Automação e Computação **Avaliação 1**
BLU3202 — **Algoritmos e Estruturas de Dados** 2019/2

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. O que será impresso pelo programa abaixo?

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

   Resposta: _____

3. 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: _____

4. O programa abaixo deve armazenar inteiros em um vetor `v` cujo tamanho é informado pelo usuário na linha 4. Informe como devem ser preenchidas as lacunas nas linhas 5, 7 e 10 para que o vetor seja criado, para que os números informados pelo usuário sejam armazenados no vetor e para que so valores lidos sejam impressos na ordem em que foram lidos.

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

   Respostas:
   – Linha 5: _____
   – Linha 7: _____
   – Linha 10: _____

5. Qual será o retorno da função `f` abaixo se os parâmetros `x` e `y` forem, respectivamente, 51 e 18?

```
1 int f(int x, int y){
2    if(y==0)
3       return x;
4    else
5       return f(y,x\%y);
6 }
```

   Resposta: _____

6. Qual será o retorno da função `f` abaixo se os `x` e `y` forem, respectivamente, 3 e 4?

```
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: _____

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 3, 11 e 14 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,
2                            int dado){
3    listaItem *novo = ___________________;
4    novo->dado = dado;
5    novo->next = NULL;
6    if(lista==NULL){
7       lista = novo;
8    }
9    else{
10       listaItem *ultimo = lista;
11       while(___________________){
12        ultimo = ultimo->next;
13       }
14       ----------------------- ;
15   }
16   return lista;
17 }
```

   Linha 3: _____
   Linha 11: _____
   Linha 14: _____

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. Seja uma matriz `m` de inteiros (`int`) de dimensão 3×3. Para acessar o conteúdo do segundo elemento da segunda linha utilizando aritmética de ponteiros, deve-se utilizar:

    (a) *((v+1)+1))
    (b) *(*(v+1)+1))
    (c) *(v+1)+1
    (d) *(v+2)+2
    (e) *((v+1)+1)