INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ

MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CURSO : Analise e desenvolvimento de dados
DISCIPLINA : Estrutura de Dados
PROFESSORA: ELANNE

Nome:_____No.:_____

Em 05.10.2023

Prova 1

1. Sobre o código abaixo é correto afirmar que: (1.0 pt)

```c
#include <stdio.h>
#define n 5
main() {
    int a[n], b[n];
    int i, j=n-1;
    for (i = 0; i < n; i++){
        printf("Valor de a[%d]: ", i);
        scanf("%d", &a[i]);
        b[j] = a[i];
        j--;
    }
    printf("\n");
    printf("VETOR B: ");
    for (i = 0; i < n; i++){
        printf("%d ", b[i]);
    }
    printf("\n\n");
}
```

a.(x) Se A for {1,2,3,4,5}, ao final do programa B será {5,4,3,2,1}
b.( ) A é um vetor de tamanho fixo e pode armazenar até 8 elementos.
c.( ) Se A for {10,9,8,7,6,5}, ao final do programa B será {5,6,7,8,9,10}
d.( ) Se A for {1,2,3,4,5}, ao final do programa B será {1,2,3,4,5}
e.( ) NDA

2. Sobre o código abaixo é correto afirmar que: (1.0 pt)

```c
#include <stdio.h>
void main() {
    int vetor[10], i, x = 0, y = 99999;
    for(i = 0; i < 10; i++) {
        printf("Digite o valor do %io. elemento: ", i);
        scanf("%d", &vetor[i]);
    }
    for(i = 0; i < 10; i++) {
        printf("%d ", vetor[i]);
        if(y > vetor[i]) {
            y = vetor[i];
        }
        if(x < vetor[i]) {
            x = vetor[i];
        }
    }
    printf("x: %d\n", x);
    printf("y: %d", y);
}
```

a.( ) Ao final do programa serão impressos o maior valor dentro da estrutura "vetor" e a posição do maior valor dentro desta estrutura.
b.(X) Ao final do programa serão impressos o maior e o menor valor dentro da estrutura "vetor".
c.( ) "vetor" é um vetor de tamanho variável que pode armazenar até 10 elementos.
d.( ) Ao final do programa serão impressos a posição do maior valor dentro da estrutura "vetor" e a posição do menor valor dentro da mesma estrutura.
e.( ) NDA

3. Sobre o código abaixo é correto afirmar: (1.0 pt)

```
#include <stdio.h>
#include <iostream>
using namespace std;
main(){
    int *w=(int*)malloc(3*sizeof(int));

    for(int i=0;i<3;i++){
        cout<<"digite o valor ["<<i<<"]:";
        cin>>w[i];
    }
    for(int i=0;i<3;i++){
        cout<<w[i]<<endl;
    }
}
```

a.( ) w é um vetor dinâmico, ou seja, a alocação sequencial é feita dinamicamente. E o programa está lendo 2 valores e armazenando os valores nas 2 (duas) primeiras posições do vetor "w".
b.(x) w é um vetor dinâmico, ou seja, a alocação sequencial é feita dinamicamente. E o programa está lendo 3 valores e armazenando em 3 posições do vetor "w".
c.( ) w é um vetor estático, ou seja, a alocação sequencial é feita estaticamente. E o programa está lendo 3 valores e armazenando em 3 posições do vetor "w".
d.( ) w é um vetor estático, ou seja, a alocação sequencial é feita estaticamente. E o programa está lendo 2 valores e armazenando em 2 posições do vetor "w".
e.( ) NDA

4. Sobre o código abaixo é correto afirmar: (1.0 pt)

```
#include <stdio.h>
#include <iostream>
using namespace std;
main(){
   int v[3]={9,6,7};
   int *p=v;
}
```

a.(x) Para imprimir o ultimo elemento de "v" podemos usar o comando: cout<<*(p+2);
b.( ) Para imprimir o ultimo elemento de "v" podemos usar o comando: cout<<p[3];
c.( ) Para imprimir o primeiro elemento de "v" podemos usar o comando: cout<<(p+1);
d.( ) Para imprimir o primeiro elemento de "v" podemos usar o comando: cout<<p;
e.( ) NDA

5. Sobre as afirmações abaixo: (1.0 pt)

A) Variáveis dinâmicas podem ser criadas pela função *malloc()* em C. No exemplo "int *p = malloc(sizeof(int));", p é um vetor alocado dinamicamente com 3 posições.

B) Variáveis dinâmicas podem ser criadas pela função *malloc()* em C. No exemplo "int *p = malloc(sizeof(int));", p é um ponteiro para inteiro e é alocado dinamicamente.

C) Um vetor em C é um mecanismo que agrega vários itens de tipos DISTINTOS, formando uma coleção HETEROGÊNEA. Exemplo: int v[3]={18,"carlos",7.5}

D) Uma *struct* em C é um mecanismo que agrega vários itens de tipos DISTINTOS, formando uma coleção heterogênea. Exemplo: typedef struct aluno {int mat; char nome[20], float nota;}

E) Considerando o trecho de código "int a=4; int *p=&a; *p=10; cout<<a;". O valor de "a" impresso ao final do trecho será de 4.

F) Considerando o trecho de código "int a=5; int *p=&a; *p=12; cout<<a;". O valor de "a" impresso ao final do trecho será de 12.

São verdadeiras as afirmações:

a.(X) B – D – F
b.( ) B – D – E
c.( ) A – D – E
d.( ) A – D – F
e.( ) NDA

6. Sobre o código abaixo é correto afirmar que: (1.0 pt)

```
main(){
    char s[3]="um";
    char t[3]="um";
    if (s==t)
       cout<<"Iguais!";
    else
       cout<<"Diferentes!";

}
```

a.(x) Será impresso a mensagem "diferentes", pois as variáveis "s" e "t", na linha “if (s==t)”, são tratadas como ponteiros que guardam endereços e por isso têm valores diferentes.
b.( ) Será impresso a mensagem "iguais", pois as variáveis que representam os vetores "s" e "t" armazenam valores iguais.
c.( ) Será impresso a mensagem "iguais", pois as variáveis "s" e "t", na linha “if (s==t)”, são tratadas como ponteiros que guardam endereços e por isso têm valores iguais.
d.( ) NDA