

QUESTÃO 1: [2 pontos] Escreva um programa para imprimir a soma de todos os números ímpares entre 0 e 600 que não são múltiplos de 3.

QUESTÃO 2: [2 pontos] Escreva um programa para imprimir os cem primeiros múltiplos positivos de um número inteiro positivo digitado pelo usuário. Exemplo: se o usuário digitar 5, o programa deve imprimir 5 10 15 20 25 30 35 etc. até 500.

QUESTÃO 3: [1 ponto] Qual será a saída apresentada pelo programa abaixo?

```
#include <stdio.h>

int minha_funcao(int x[3]) {
    int y = x[0] + x[1]++;
    return y+2;
}

int main() {
    int num[3] = {10, 20, 30};
    printf("\nresultado = %d", minha_funcao(num));
    printf("\nsoma = %d", num[0] + num[1] + num[2]);
    return 0;
}
```

QUESTÃO 4: [2,5 pontos] Olhe *com muita atenção* os trechos de programa abaixo, e responda o que será impresso na tela quando cada um dos trechos é executado:

```
a) typedef struct {
    int x;
    int y;
} PONTO;

PONTO p, q;
PONTO* ptr_ponto;
p.x = 3;
p.y = 4;
q.x = 13;
q.y = 14;
ptr_ponto = &p;
(*ptr_ponto).x++;
ptr_ponto = &q;
ptr_ponto->y++;
printf("%d %d", p.x + p.y, q.x + q.y);
```

- b)

```
int y = 10;
int* ptr = &y;
*ptr = 12;
ptr = malloc(sizeof(int));
*ptr = 13;
printf("%d %d", y, *ptr);
```
- c)

```
int i;
int* ptr;
ptr = (int*) malloc(3 * sizeof(int));
for (i = 0; i < 3; i++) {
    ptr[i] = 2*i;
}
printf("%d %d", *(ptr+1), *(ptr+2));
```
- d)

```
char bicho[10] = "cachorro";
char animal[10];
strcpy(animal, bicho);
animal[2] = 'o'; // letra o
animal[3] = 0; // valor zero (de terminacao)
printf("%s %s", bicho, animal);
```
- e)

```
int array[3] = {10, 20, 30};
int* ptr;
ptr = array;
array[0] = 21;
printf("%d %d", array[0], *ptr);
```

QUESTÃO 5: [2,5 pontos] Faça um programa como descrito a seguir.

a) Crie uma estrutura *ESTRADA* para armazenar as características de uma estrada: *cidade_origem* (array estático para 30 caracteres), *cidade_destino* (array estático para 30 caracteres), *distancia* (int) e *valor_pedagio* (float).

b) Crie uma função *calcular_pedagio* que recebe como parâmetros um array dinâmico para o tipo *ESTRADA* e a quantidade de elementos armazenados no array, e retorna a soma dos valores dos pedágios de todas as estradas que estão no array.

c) No *main()*, peça para o usuário digitar o número de estradas de uma região, declare e aloque memória para um array dinâmico que possa guardar aquela quantidade digitada pelo usuário de elementos *ESTRADA*.

d) Ainda no *main()*, peça para o usuário preencher todos os campos de cada uma das estradas do array do item (c).

e) Ainda no *main()*, e utilizando a função criada em (b), imprima o valor total dos pedágios das estradas digitadas pelo usuário no item (d).