- 1. Em relação ao setor de memória na Figura a seguir (um fictício com endereços de 16 bits):
 - a. Qual o endereço base de 3.2 e qual o do número inteiro 503?
 - b. O que é escrito na tela com printf("%s\n", p0);?
 - c. O que é escrito na tela com printf("%s\n", &p0[5]);?
 - d. O que é escrito na tela com printf(" $%s\n"$, p0+2);?
 - e. Qual o valor de *p1?
 - f. Qual o valor de p1[0]?
 - g. Qual o valor de p1+1?
 - h. Qual o valor de *(p1+1)?
 - i. Qual o valor de *(p2+1)?
 - j. Qual o valor de &x?
 - k. Qual o valor de &p0?
- 2. Altere o ponteiro p para apontar para o início da segunda palavra da string. Assuma que a frase digitada possui ao menos uma palavra.

```
int main() {
    char frase[201];
    fgets(frase, 201, stdin);
    char *p = frase;
    //altere p para que aponte para o primeiro
    //caractere apos o primeiro espaco
    printf("%s\n", p);
    return 0;
}
```

3. Altere o valor inicial de p
 de forma que a saída do programa seja 5 11 12 2 8.

```
#include <stdio.h>
int main() {
    int i;
    int v[] = {3, 14, 9, 6, 5, 11, 12, 2, 8, 13, 7, 10, 1, 4};
    int *p; //atribua um valor inicial adequado

    for(i = 0; i < 5; i++) {
        printf("%d ", p[i]);
    }
    return 0;
}</pre>
```

- 4. ⊳ Em relação ao setor de memória da Figura da primeira questão:
 - a. Qual o endereço base de 5.4 e qual o do número inteiro 702?
 - b. Considerando p0 = 0x85bc, o que é escrito na tela com printf("%s\n", p0);?
 - c. Considerando p0 = 0x85bc, o que é escrito na tela com printf("%s\n", &p0[4]);?
 - d. Considerando p0 = 0x85bc, o que é escrito na tela com printf("%\n", p0+1);?
 - e. Qual o valor de *p2?

| 0x85bb 0 1 1 0 0 0 1 1 char c 0x85bc 0 1 1 0 0 0 1 1 char a 0x85bd 0 1 1 0 0 0 0 0 1 char a 0x85bd 0 1 1 0 0 0 0 0 char a 0x85bd 0 1 1 0 0 0 0 char a 0x85c0 0 1 1 0 1 0 0 char i 0x85c1 0 1 1 0 0 0 0 char i 0x85c2 0 0 0 0 0 0 char i 0x85c3 1 1 0 0 1 0 0 0x85c4 1 1 0 0 1 1 0 0 <th></th> <th>_</th> <th></th> <th>_</th> <th></th> <th></th> <th></th> <th></th> <th>_</th> <th></th> | | _ | | _ | | | | | _ | |
|---|------------------------------|-----|---|---|---|---|-------|--------|---|--------|
| 0x85bc | char c | 700 | 1 | 0 | 0 | 0 | 17000 | 100000 | 0 | |
| 0x85bd 0 | char a | 1 | - | 0 | 0 | 0 | 1 | 1 | 0 | 0x85bb |
| 0x85be 0 0 0 0 0 0 0 char \(\)0 0x85bf 0 1 1 0 0 0 1 char a 0x85c0 0 1 1 0 1 1 0 char 1 0x85c1 0 1 1 0 1 0 0 char \(\)0 0x85c2 0 0 0 0 0 0 char \(\)0 0x85c3 0 1 0 0 1 0 0 0x85c4 0 1 0 0 1 1 0 0 0x85c5 1 1 0 0 1 1 0 0 0x85c6 1 1 0 0 1 1 0 1 0x85c7 1 1 0 1 1 0 1 1 0 0 0x85c6 1 | $\mathbf{char} \ \mathbf{s}$ | - | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0x85bc |
| 0x85bf | char a | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| $\begin{array}{c ccccccccccccccccccccccccccccccccccc$ | char $\setminus 0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85be |
| 0x85c1 0 1 1 0 1 char 0 0x85c2 0 0 0 0 0 0 char 0 0x85c3 0 1 0 0 0 0 0 0 0 0x85c4 0 1 0 0 1 1 0 | char a | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0x85bf |
| 0x85c2 0 <td>char l</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0x85c0</td> | char l | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0x85c0 |
| 0x85c3 0 1 0 0 0 0 0 float 3.2 0x85c4 0 1 0 0 1 1 0 0 0x85c5 1 1 0 0 1 1 0 0 0x85c6 1 1 0 0 0 0 0 0 0x85c7 0 1 0 0 0 0 0 0 0x85c8 1 0 1 1 0 0 1 1 0 0 0x85c9 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 0 0 <td>char i</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0x85c1</td> | char i | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0x85c1 |
| $\begin{array}{c ccccccccccccccccccccccccccccccccccc$ | char $\setminus 0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85c2 |
| 0x85c6 1 1 0 0 1 1 0 0 0x85c7 0 1 0 0 0 0 0 0 0 0x85c8 1 0 1 0 1 1 0 | float 3.2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0x85c3 |
| 0x85c6 1 1 0 0 1 0 <td></td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0x85c4</td> | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0x85c4 |
| 0x85c7 0 1 0 0 0 0 0 0 float 5.4 0x85c8 1 0 1 0 1 1 0 0 0x85c9 1 1 0 0 1 1 0 0 0x85ca 1 1 0 0 1 1 0 1 0x85cb 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 1 0 | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0x85c5 |
| 0x85c9 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 <td></td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0x85c6</td> | | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0x85c6 |
| 0x85c9 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 <td>float 5.4</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0x85c7</td> | float 5.4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0x85c7 |
| 0x85ca 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 <td></td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0x85c8</td> | | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0x85c8 |
| 0x85cb 0 0 1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 <td></td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0x85c9</td> | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0x85c9 |
| 0x85cc 1 0 0 1 1 0 0 1 0x85cd 1 0 0 1 1 0 0 1 0x85ce 1 0 0 1 1 0 1 0 0x85cf 0 1 0 0 0 0 1 1 0 0x85d0 0 0 0 0 1 1 0 0 1 1 0 0x85d1 0 1 1 0 0 1 1 0 0 1 1 0 0x85d2 0 1 1 0 0 1 0 1 0 | | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0x85ca |
| 0x85cd 1 0 0 1 1 0 0 1 0x85ce 1 0 0 1 0 1 0 1 0 0x85cf 0 1 0 0 0 0 1 1 0 0x85d0 0 0 0 0 1 1 0 0 1 1 0 0x85d1 0 1 1 0 0 1 1 0 0 1 1 0 0x85d2 0 1 1 0 0 1 0 <t< td=""><td>float 1.2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0x85cb</td></t<> | float 1.2 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0x85cb |
| 0x85ce 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 float 8.4 0x85d0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0x85cc |
| 0x85cf 0 1 0 0 0 0 1 float 8.4 0x85d0 0 0 0 0 1 1 0 0x85d1 0 1 1 0 0 1 1 0 0x85d2 0 1 1 0 0 1 1 0 0x85d3 1 0 0 0 0 1 0 1 char *p0 = 0x85ba 0x85d4 1 0 1 1 0 1 0 1 0 | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0x85cd |
| 0x85d0 0 0 0 0 1 1 0 0x85d1 0 1 1 0 0 1 1 0 0x85d2 0 1 1 0 0 1 1 0 0x85d3 1 0 0 0 0 1 0 1 char *p0 = 0x85ba 0x85d4 1 0 1 1 0 1 0 1 char *p0 = 0x85ba 0x85d5 0 <td></td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0x85ce</td> | | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0x85ce |
| 0x85d1 0 1 1 0 0 1 1 0 0x85d2 0 1 1 0 0 1 1 0 0x85d3 1 0 0 0 0 1 0 1 char *p0 = 0x85ba 0x85d4 1 0 1 1 0 1 0 0x85d5 0 0 0 0 0 0 0 0 0x85d6 0 0 0 0 0 0 0 0 0x85d7 0 0 0 0 0 0 0 0 0x85d8 1 1 1 1 0 1 1 1 0 0 0 0 char d 0x85d8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | float 8.4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0x85cf |
| 0x85d2 0 1 1 0 0 1 1 0 0x85d3 1 0 0 0 1 0 1 char *p0 = 0x85ba 0x85d4 1 0 1 1 1 0 1 0 0x85d5 0 | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0x85d0 |
| 0x85d3 1 0 0 0 1 0 1 char *p0 = 0x85ba 0x85d4 1 0 1 1 0 1 0 0x85d5 0 0 0 0 0 0 0 int x = 503 0x85d6 0 0 0 0 0 0 0 0x85d7 0 0 0 0 0 0 0 0 0x85d8 1 1 1 1 0 1 1 1 1 0 0 1 0 | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0x85d1 |
| 0x85d4 1 0 1 1 1 0 1 0 1 0 1 0 1 0 <td></td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0x85d2</td> | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0x85d2 |
| 0x85d5 0 <td>char *p0 = 0x85ba</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0x85d3</td> | char *p0 = 0x85ba | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0x85d3 |
| 0x85d6 0 <td></td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0x85d4</td> | | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0x85d4 |
| 0x85d7 0 0 0 0 0 0 1 0x85d8 1 1 1 1 0 1 1 1 0x85d9 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 1 0 0 0 0 | int $x = 503$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85d5 |
| 0x85d8 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 <td></td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0x85d6</td> | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85d6 |
| 0x85d9 0 1 1 0 0 1 0 0 char d 0x85da 0 0 0 0 0 0 0 0 int 702 0x85db 0 0 0 0 0 0 0 0 0x85dc 0 0 0 0 0 1 0 0x85dd 1 0 0 0 1 0 1 0 0x85df 1 1 0 0 0 1 1 1 0x85e0 1 0 0 0 1 0 1 1 0x85e1 1 1 0 0 1 0 1 1 | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85d7 |
| 0x85da 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 <td></td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0x85d8</td> | | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0x85d8 |
| 0x85db 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 <td>char d</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0x85d9</td> | char d | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0x85d9 |
| 0x85db 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 <td>int 702</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0x85da</td> | int 702 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85da |
| 0x85dd 1 0 1 1 1 1 0 0x85de 1 0 0 0 1 0 1 float *p1 = 0x85c3 0x85df 1 1 0 0 0 1 1 0x85e0 1 0 0 0 1 0 1 float *p2 = 0x85cb 0x85e1 1 1 0 0 1 1 1 | nous per la la constante | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0x85de 1 0 0 0 0 1 0 1 float *p1 = 0x85c3 0x85df 1 1 0 0 0 1 1 0x85e0 1 0 0 0 1 0 1 float *p2 = 0x85cb 0x85e1 1 1 0 0 1 0 1 1 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0x85dc |
| 0x85de 1 0 0 0 0 1 0 1 float *p1 = 0x85c3 0x85df 1 1 0 0 0 1 1 0x85e0 1 0 0 0 1 0 1 float *p2 = 0x85cb 0x85e1 1 1 0 0 1 0 1 1 | | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0x85dd |
| 0x85df 1 1 0 0 0 0 1 1 0x85e0 1 0 0 0 1 0 1 float *p2 = 0x85cb 0x85e1 1 1 0 0 1 0 1 1 | float *p1 = $0x85c3$ | 1 | 0 | 1 | _ | | 0 | 0 | 1 | |
| 0x85e0 | 10 - 20 | 1 | 1 | 0 | 0 | 0 | 0 | | 1 | |
| 0x85e1 1 1 0 0 1 0 1 1 | float *p2 = 0x85cb | | | | | 0 | 7.0 | | | |
| | 100 /s | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | |
| 0x6562 1 0 0 0 1 0 1 Int p3 = 0x850a | int *p3 = 0x85ca | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0x85e2 |
| 0x85e3 1 1 0 0 1 0 1 0 | • | 0 | | | | 0 | | | | |

Figure 1: Setor da memória para as questões 1 e 4

- f. Qual o valor de p2[0]?
- g. Qual o valor de p2+1?
- h. Qual o valor de (p1+2)?
- i. Qual o valor de *(p1+3)?
- j. Qual o valor de &d?
- k. Qual o valor de &p2?
- 5. Implemente uma função soma Diferenca que retorna indiretamente, através de ponteiros, a soma e a diferença de dois inteiros a e b:

```
void somaDiferenca(int a, int b, int *soma, int *diferenca)
```

A função main deve ler dois inteiros \mathbf{x} e \mathbf{y} e escrever na tela a soma e a diferença desses dois números através do uso da função soma Diferença.

Exemplos:

| Input | Output |
|-------|--------|
| 2 3 | 5 -1 |
| 5 4 | 9 1 |
| 12 9 | 21 3 |

6. > Criar função para retornar indiretamente a diferença entre duas coordenadas

```
void diferenca(int px, int py, int qx, int qy, int *dx, int *dy)
```

7. Escreva uma função para retornar uma nova string, igual à passada como parâmetro, mas em caixa-alta:

```
char * caixaAlta(char *s)
```

A função main deve ler do usuário uma frase de até 100 caractees e escrever na tela a string em caixa-alta fazendo uso da função caixaAlta. Não esqueça de liberar a memória após o uso.

Exemplos:

| Input | Output |
|------------------|------------------|
| Este eh um teste | ESTE EH UM TESTE |
| MaIS outro TESTE | MAIS OUTRO TESTE |

8. Crie uma função **misturar** que recebe 2 strings como parâmetros e retorna uma nova string (alocada dinamicamente) alternando as letras de ambas as strings. Caso não possuam o mesmo tamanho, as letras restantes da string maior devem ser colocadas ao final da nova string. Por exemplo, ao misturar "teste" e "algo", obtém-se "taelsgtoe". Já ao misturar "asa" e "inconstitucionalidade" obtém-se "aisnaconstitucionalidade".

```
char * misturar(char *str1, char *str2)
```

A função main deve ler duas palavras de até 100 caracteres, chamar a função misturar e escrever na tela o resultado. Não se esqueça de liberar a memória do que foi alocado dinamicamente.

| Input | Output |
|---------------------------|--------------------------|
| teste algo | taelsgtoe |
| asa inconstitucionalidade | aisnaconstitucionalidade |
| rslao eutd | resultado |

9. \triangleright Crie uma função que retorna uma nova string igual à passada como parâmetro mas substituindo todos os caracteres \mathbf{x} pelo caractere \mathbf{y} :

```
char * substituir(char *s, char x, char y);
```

Por exemplo, ao chamar substituir(palavra, 'a', 'e'), sendo palavra lata, retorna lete.

Escreva a função main da seguinte forma, programando o que é solicitado nos comentários:

```
int main() {
    char p[51], x, y, *nova;
    fgets(p, 51, stdin);
    if(p[strlen(p)-1] == '\n')
        p[strlen(p)-1] = '\0';
    scanf("%c %c", &x, &y);

    //obter a nova string chamando a função substituir

    //escrever na tela a string nova e a string original
    //liberar memória da string nova
    return 0;
}
```

Exemplo:

| Input | Output |
|----------|-------------------|
| um aviao | um evieo um aviao |
| a e | |

10. Escreva a seguinte função para dobrar o inteiro cujo endereço está sendo passado como parâmetro:

```
void dobrarInteiro(int *a);
```

Na função main, leia um inteiro, chame a função e escreva na tela o novo valor do inteiro.

Exemplo:

| Input | Output |
|-------|--------|
| 3 | 6 |

11. ⊳ Escreva a seguinte função para colocar dois inteiros, cujos endereços estão sendo passados como parâmetros, em ordem crescente:

```
void ordenarDoisInteiros(int *a, int *b);
```

Na função main, leia um inteiro a, leia um inteiro b, chame a função e escreva na tela os valores de a e b.

Exemplo:

| Input | Output |
|-------|--------|
| 3 2 | 2 3 |
| 8 11 | 8 11 |
| 9 3 | 3 9 |
| 1 2 | 1 2 |

12. Escreva um programa que leia um inteiro \mathbf{n} , leia n números reais e escreva na tela o índice (começando de 1) do maior entre esses \mathbf{n} números reais. Assuma que não há números iguais na sequência.

Exemplo:

| Input | Output |
|---|--------|
| 6 3.97 2.15 13.97 12.38 10.65 16.19 | 6 |

13. ⊳ O MEC precisa de sua ajuda para automatizar a correção das provas objetivas do ENEM! Escreva um programa que leia um número inteiro **n** representando o número de questões (não há limite para o número de questões). Em seguida leia as **n** respostas do gabarito e, em seguida, as **n** respostas do aluno. Assuma que as respostas estão sempre entre 1 e 5. Depois o programa deve escrever na tela quantas questões o aluno acertou e a string "acertos" ou "acerto" (para 1 acerto), conforme exemplo abaixo.

Exemplos:

| Input | Output |
|-----------------------|-----------|
| 4 | 2 acertos |
| 1 2 3 4 | |
| 1 5 3 5 | |
| 7 | 1 acerto |
| $1\ 2\ 3\ 2\ 1\ 5\ 4$ | |
| 3 3 3 3 3 3 3 | |

14. Em seus testes, a partir de qual valor n o código seguinte gera erro de execução? Altere o programa para possibilitar que o programa execute com um valor de n maior.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
   int n;
   scanf("%d", &n);
   int v[n];
   for(int i = 0; i < n; i++)
      v[i] = (i*i)%10;
   int soma = 0;</pre>
```