

```

1. main() {
    int myCount = 0;
    while (myCount < 10)
    {
        printf("%d", myCount + 1);
    }
}

```

e) Entrará em loop infinito, pois myCount será sempre 0, não é incrementado.

2. main() { Trecho 1

```

    int mat[2][2] = {{1, 2}, {3, 4}};
    int i, j;
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            printf("%d\n", mat[i][j]);
    system("pause");
}

```

R: 1 2
3 4

1	2
3	4

- Matriz estática, printa os valores da 1ª linha (col 1 e 2) e depois da 2ª linha (col 1 e 2)

main() { Trecho 2

```

    int mat[2][2] = {{1, 2}, {3, 4}};
    int *p = &mat[0][0];
    int i;
    for (i = 0; i < 4; i++)
        printf("%d\n", *(p+i));
    system("pause");
}

```

R: 1
2
3
4

* (p) * (pd)

1	2
3	4

- Obtém o mesmo resultado que o Trecho 1, porém usa um ponteiro que aponta pra endereço inicial da matriz e printa o valor apontado a partir do end. inicial. Aponta na 1ª linha, percorre ela através dos prints até o final da 2ª linha

3. main() {

```

    int i = 3, j = 5;
    int *p, *q;
    p = &i;
    q = &j;
}

```

a) $p = \&i$ p recebe o endereço de i

b) $*p - *q = 3 - 5 = -2$

c) $**\&p = 3$

d) $3 * (- *p / (*q) + 7) = 6$

4. τ ocupa o end 1000
main() {

int i = 5, *p;

p = &i;

printf("%p, %d %d %d %d\n", p, *p+2, **&p, 3**p, **&p+4);

Snima 1000, 7, 5, 15, 9

5. R. (a) $*(pulo+2)$

6. int mat[4], *p, x;

a) $p = mat + 1$; V, p recebe o endereço da posição inicial do vetor + 1 (2ª posição)

b) $p = mat++$; N V, pois mat é um vetor e a sintaxe está errada

c) $p = ++mat$; N V, pois mat é um vetor e a sintaxe está errada

d) $x = (*mat)++$; V, x recebe o valor apontado na posição inicial da matriz e depois faz a incrementação deste no valor apontado naquela posição

7. Cód.

8. Cód

9. char *Teste(char *s1, const char *s2) { s1 = Livro 18

char *aux = s1;

while(*s1) s1++; \rightarrow PARA NA POSIÇÃO DO \0

for(; *s1 = *s2) != '\0'; s1++, s2++; s1 = Livro 19

return aux;

}

Resposta é

passa a

string s2

por s1 na

região apontada

por s1 no

momento

que é onde

tem o '\0'

s2: curso 10

10. `int F1(unsigned int n) {`

`if(n==0) return n;`

`int i, j;`

`for(i=j=1; i<2*n-1; i+=2, j+=i),`

`return j;`

`}`

$n=10$

1	1	2
3	4	11
5	9	26
7	16	

a) Função que dado um número N , armazena em j todos os números de ≥ 0 e $< N$ elevado ao quadrado e no final retorna N^2 . E se $N=0$ retorna 0.

b) $j=100$

c) `int F1(unsigned int n) {`

`if(n==0) return n;`

`int i, j;`

`for(i=1; i<2*n-1; i+=2) {`

`for(j=1; j<N^2-1; j+=i),`

`}`

`}`