

# Questão 1.

Resposta da questão II & III

Minha resposta II.

## Pergunta 1

0 / 0,55 pts

Um ponteiro ou apontador é uma variável que armazena um endereço de memória. Em relação aos ponteiros, avalie as afirmações a seguir.

I. Referente a alocação dinâmica de memória em C, a função `clear` é usada para limpar o conteúdo de um ponteiro.

II. Após a sequência de instruções abaixo, se executarmos a instrução `"p = 7;"`, `x` passará a ter o valor 7.

```
int *p;  
int x = 5;  
p = &x;
```

III. A saída na tela abaixo será 24

```
int soma(int *a, int *b){  
    *a = *a + *b;  
    return *a;  
}  
int main(){  
    int x = 5, y = 3;  
    x = 5; y = 3;  
    y = soma(&x, &y);  
    printf("soma: %d\n", (x + y));  
    return 0;  
}
```

É correto o que se afirma em

☐ I, II e III.

☒ II, apenas.

☐ II e III, apenas.

I.(Falsa) Clear não é usado e sim o Free;

II.(Verdadeira) Código Compilado no Repl it

<pre>int main(void) {     int *p;     int x = 5 ;     p = &amp;x;     *p =7;     printf("%d",x); }</pre>	
--	--

III.(Falso) Código Compilado no Repl it

Resposta não é igual a 24 e sim 16.

```
#include <stdio.h>  
  
int soma(int *a, int*b){  
    *a = *a+b;  
    return *a;  
}  
  
int main(void) {  
    int x = 5 , y = 3;  
    y = soma(&x , &y);  
    printf("%d + %d = %d /n",x,y,(x+y));  
}
```

8 + 8 = 16 /n

## Questão 2.

Resposta da questão II

Minha resposta na prova I (por falta de alternativa, pois todas estão erradas).

Um ponteiro é uma variável capaz de armazenar um endereço de memória ou o endereço de outra variável. Em relação aos ponteiros, avalie as afirmações a seguir.

I. Após a sequência de instruções abaixo, p é igual a 5.

```
int *p;  
int x = 5;  
p = &x;
```

II. A saída na tela abaixo será 13

```
int soma(int *a, int *b){  
    *a = *a + *b;  
    return *a;  
}  
int main(){  
    int x = 5, y = 3;  
    x = 5; y = 3;  
    y = soma(&x, &y);  
    printf("soma: %d\n", (x + y));  
    return 0;  
}
```

III. Uma forma de alocarmos um vetor de números reais com dez posições na linguagem C é fazendo `double* vet = (double*) malloc(10 * sizeof(double) + 1);`

É correto o que se afirma em

☐ I, II e III.

☐ II, apenas.

☒ I, apenas.

I.(Falso),

O valor de P é igual a um endereço de memória e não 5, porem o \*P e igual ao conteúdo de x que por sua vez e 5.

```
int main(void) {  
    int *p;  
    int x = 5 ;  
    p = &x;  
    printf("%d",p);  
}
```

1578569700

## II.(Falso) Código Compilado no Repl it

Resposta não é igual a 13 e sim 16.

```
#include <stdio.h>

int soma(int *a, int*b){
    *a = *a+b;
    return *a;
}

int main(void) {
    int x = 5 , y = 3;
    y = soma(&x , &y);
    printf("%d + %d = %d /n",x,y,(x+y));
}
```

8 + 8 = 16 /n

## III.(Falso) = a alocação deve ser feita

double\* vet = (double\*) malloc(10 \* sizeof(double));