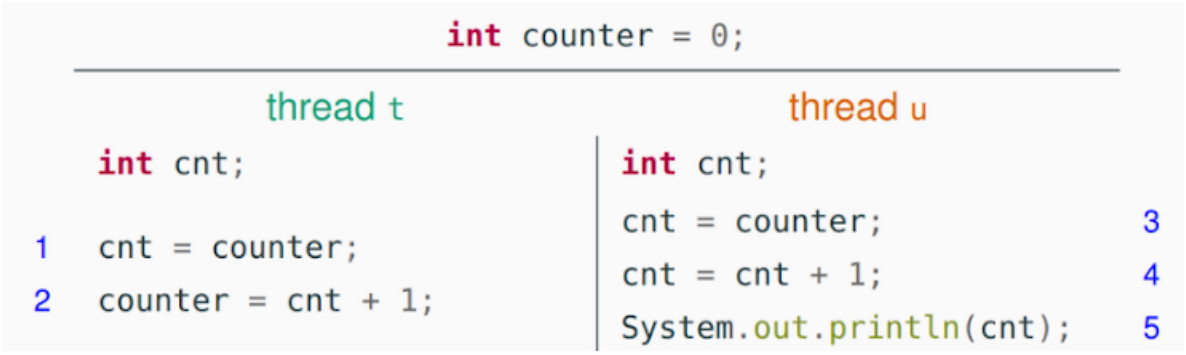
PCD - Quiz 2

O e-mail do participante (**igor.ribeiro@unifesp.br**) foi registrado durante o envio deste formulário.

O que o programa da figura imprime? (supondo operações atômicas, ou seja, cada linha de código é executada do início ao fim, sem interrupções ou troca de contexto)

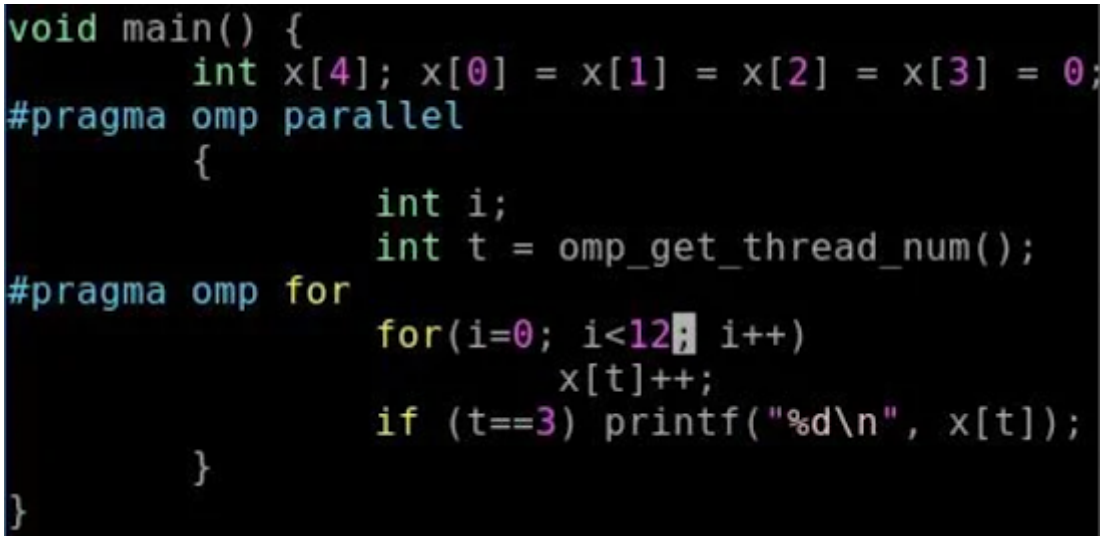


- ○ 0 (zero)
- ○ 0 ou 1
- ○ 1
- ◉ 1 ou 2

O que retorna a seguinte chamada à função: N = omp_get_thread_num() ?

- ○ Número total de threads em execução
- ○ Número de processadores
- ◉ ID da thread
- ○ N.D.A.

Definindo: OMP_NUM_THREADS=4, qual o resultado do programa da figura?



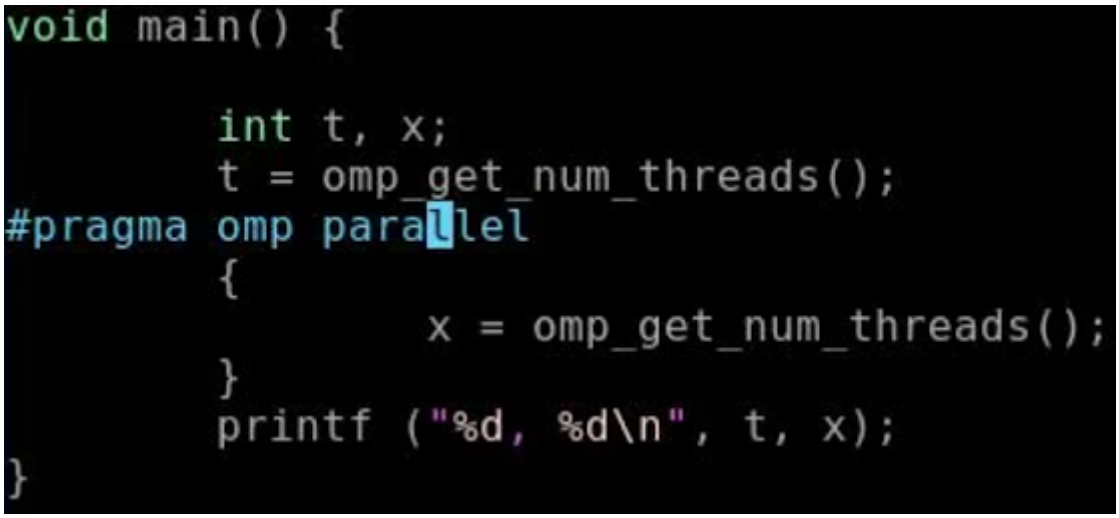
```
void main() {
        int x[4]; x[0] = x[1] = x[2] = x[3] = 0;
#pragma omp parallel
        {
                int i;
                int t = omp_get_thread_num();
#pragma omp for
                for(i=0; i<12; i++)
                        x[t]++;
                if (t==3) printf("%d\n", x[t]);
        }
}
```

- ◉ 12
- ○ 3
- ○ 0
- ○ 4

Resposta correta

- ◉ 3

Para OMP_NUM_THREADS=3, qual o resultado do programa da figura?

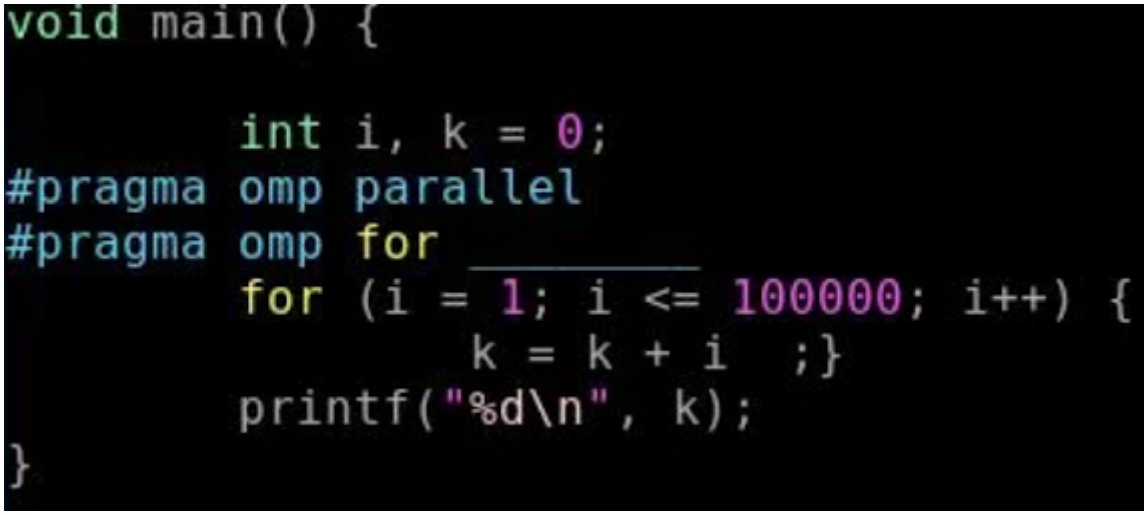


```
void main() {

        int t, x;
        t = omp_get_num_threads();
#pragma omp parallel
        {
                x = omp_get_num_threads();
        }
        printf ("%d, %d\n", t, x);
}
```

- ○ 1,1
- ○ 3,1
- ○ 3,3
- ◉ 1,3

Qual o complemento correto na linha #pragma omp for _____ no programa exibido???

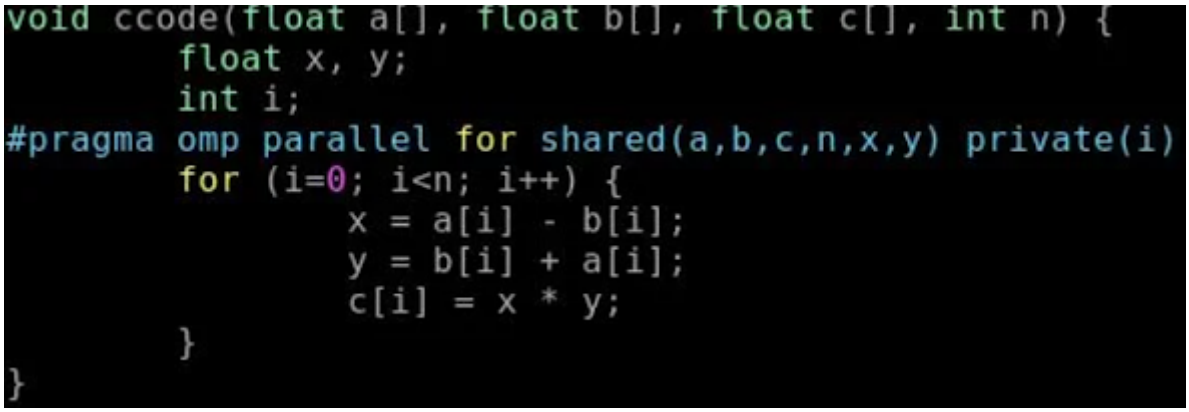


```
void main() {

        int i, k = 0;
#pragma omp parallel
#pragma omp for _________
        for (i = 1; i <= 100000; i++) {
                k = k + i  ;}
        printf("%d\n", k);
}
```

- ○ nenhum complemento
- ○ private(i) shared(k)
- ○ private(i,k)
- ◉ private(i) reduction(+:k)

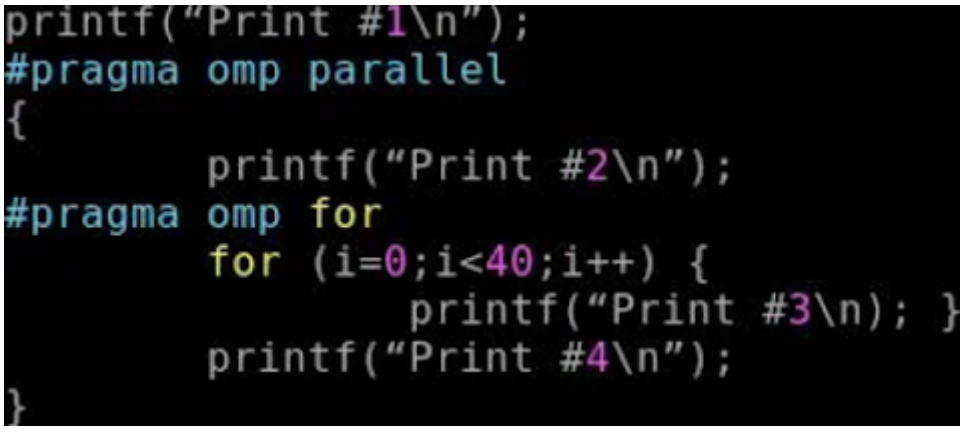
Qual o erro no codigo exibido que causa erros nos resultados esperados?



```
void ccode(float a[], float b[], float c[], int n) {
        float x, y;
        int i;
#pragma omp parallel for shared(a,b,c,n,x,y) private(i)
        for (i=0; i<n; i++) {
                x = a[i] - b[i];
                y = b[i] + a[i];
                c[i] = x * y;
        }
}
```

- ○ Falta uma operação de redução
- ○ Não há erros
- ◉ as variáveis x e y estão compartilhadas mas deveriam ser privadas
- ○ variável i deve ser compartilhada

Quantas vezes Print #1, #2, #3 3 #4 serão impressos, para 4 threads no código exibido?



```
printf("Print #1\n");
#pragma omp parallel
{
        printf("Print #2\n");
#pragma omp for
        for (i=0;i<40;i++) {
                printf("Print #3\n"); }
        printf("Print #4\n");
}
```

- ◉ #1 - 1x, #2 - 4x, #3 - 40x, #4 - 4x
- ○ #1 - 4x, #2 - 4x, #3 - 40x, #4 - 40x
- ○ #1 - 1x, #2 - 4x, #3 - 160x, #4 - 4x
- ○ #1 - 1x, #2 - 4x, #3 - 40x, #4 - 40x

Em uma região paralela em OpenMP, as variáveis, em sua maioria, são consideradas do tipo:

- ○ Locais (private)
- ○ Não existe definição padrão
- ◉ Compartilhadas (shared)
- ○ N.D.A.

Entre todos os argumentos passados para a função pthread_create, o argumento final representa?

- ○ Atributos da thread
- ○ Nome da Thread
- ◉ Representam dados passados para a Thread
- ○ Thread ID

O tipo de dado usado para armazenar uma thread é

- ○ pthread
- ◉ pthread_t
- ○ p_thread_t
- ○ pthread_id

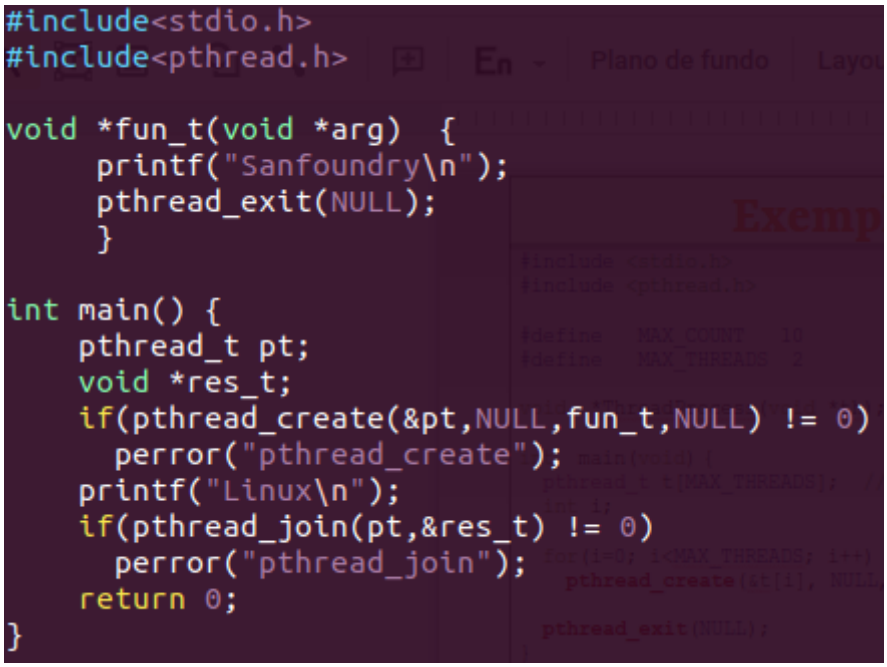
Quais das funções abaixo podem ser usadas para sincronizar a execução de múltiplas threads?

- ○ pthread_exit
- ○ pthread_cancel
- ○ pthread_self
- ◉ pthread_join

pthread_create(&threads_id, NULL, Hello_world, NULL); cria uma thread com quais características?

- ○ Chama Hello_world com argumentos do tipo padrão (default)
- ◉ Chama Hello_world com argumentos do tipo NULL

Qual string será exibida primeiro no código PThreads exibido?



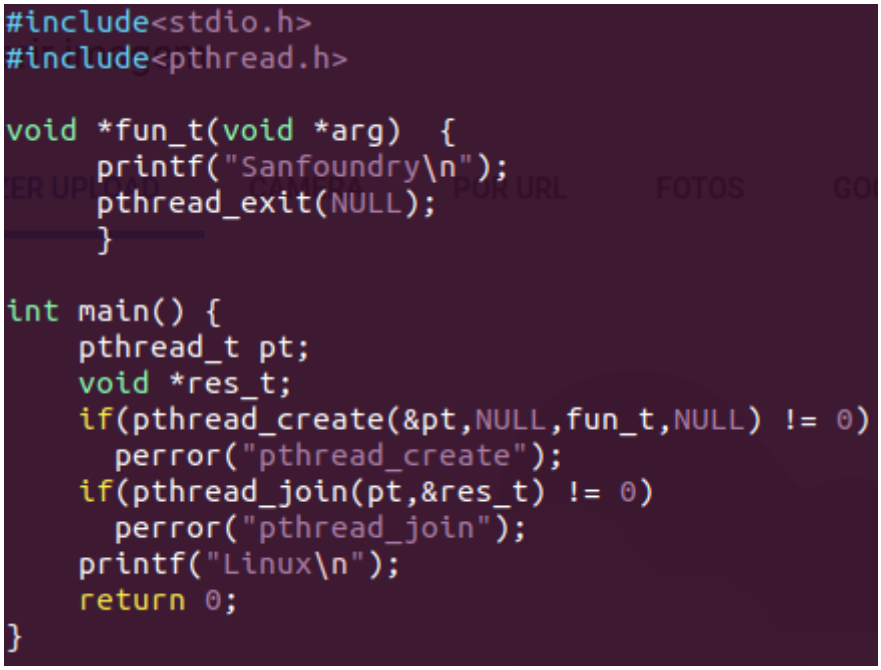
```c
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg)  {
     printf("Sanfoundry\n");
     pthread_exit(NULL);
     }

int main() {
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
      perror("pthread_create");
    printf("Linux\n");
    if(pthread_join(pt,&res_t) != 0)
      perror("pthread_join");
    return 0;
}
```

- ○ Linux
- ○ Sanfoundry
- ○ Não executa e gera erro de compilação
- ◉ Impossível predizer, pois depende do escalonador.

Qual string será exibida primeiro no código PThreads exibido (atenção: código diferente do caso anterior)?



```c
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg)  {
     printf("Sanfoundry\n");
     pthread_exit(NULL);
     }

int main() {
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
      perror("pthread_create");
    if(pthread_join(pt,&res_t) != 0)
      perror("pthread_join");
    printf("Linux\n");
    return 0;
}
```

- ○ Linux
- ◉ Sanfoundry
- ○ Não executa e gera erro de compilação
- ○ Impossível predizer, pois depende do escalonador.

O que será exibido na tela como saída do programa da figura?

```c
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg)  {
    printf("Sanfoundry\n");
    pthread_exit(NULL);
    }

int main() {
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
      perror("pthread_create");
    return 0;
}
```

- (●) Provavelmente nada, pois o programa principal termina antes da thread executar
- ( ) Sanfoundry
- ( ) Não executa e gera erro de compilação
- ( ) N.D.A.

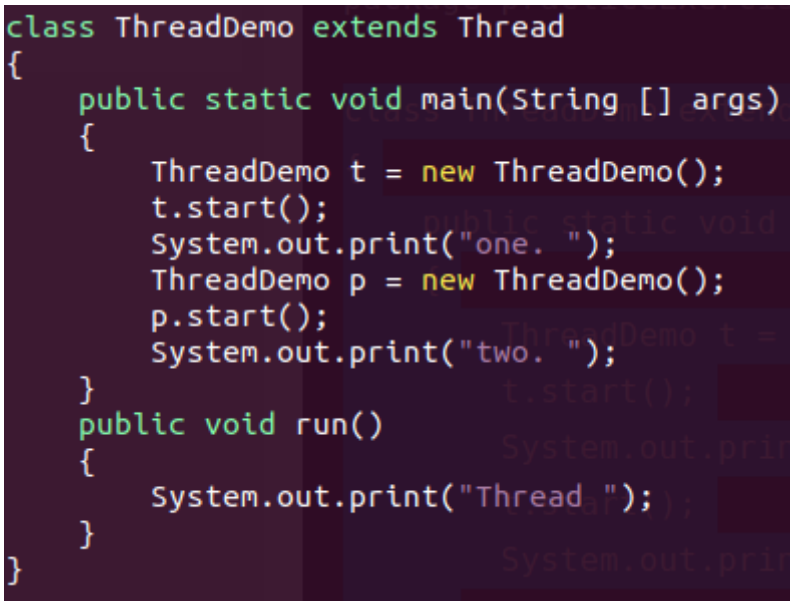
Como se deve fazer para instanciar e executar o objeto como uma thread no programa anexo?

```
class Demo implements Runnable
{
    public void run() {
     System.out.println(“Thread is in Running state”);
     }

    public static void main(String args[])
    {
        /* Missing code? */
    }
    }
```

- ○ Thread t = new Thread(X); t.start();
- ○ Thread t = new Thread(); t.run();
- ◉ Demo X = new Demo(); new Thread(X).start();
- ○ N.D.A.

O que será exibido no programa JavaThreads da figura?



```
class ThreadDemo extends Thread
{
    public static void main(String [] args)
    {
        ThreadDemo t = new ThreadDemo();
        t.start();
        System.out.print("one. ");
        ThreadDemo p = new ThreadDemo();
        p.start();
        System.out.print("two. ");
    }
    public void run()
    {
        System.out.print("Thread ");
    }
}
```

- ○ Thread one. Thread two.
- ○ one. Thread two. Thread
- ○ one. two. Thread Thread
- ◉ Impossivel determinar, pois se configura uma condição de corrida.