

1. Find outputs of the following code.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
```

```
void *t_func(void *arg);
int var=0;
int t_id[]={1,2};
int main(){
```

```
    pthread_t t1;
    pthread_t t2;
    int a1[]={t_id[0],5};
    int a2[]={t_id[1],3};
    pthread_create(&t1,NULL,t_func,(void *)a1);
    pthread_join(t1,NULL);
    pthread_create(&t2,NULL,t_func,(void *)a2);
    pthread_join(t2,NULL);
    printf("Value of var after operations of threads: %d\n",var);

    return 0;
```

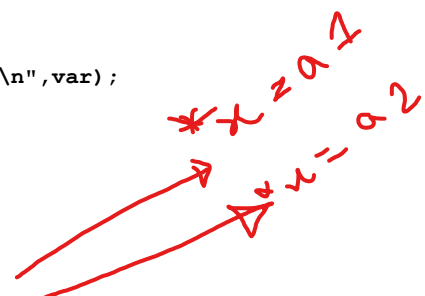
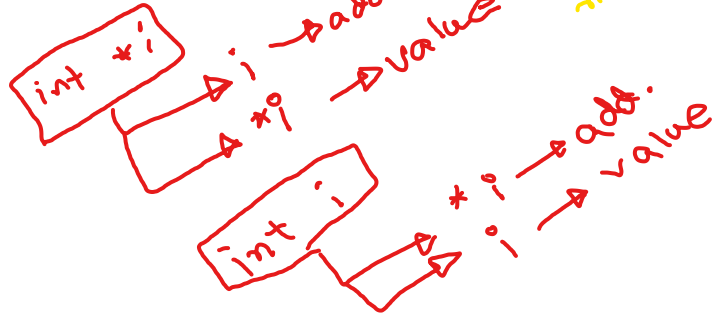
```
void *t_func(void *arg){
    int *x=arg;
    if(x[0]==1){
        printf("Entered in Thread :%d\n",x[0]);
        var+=x[1];
        printf("Value of var after the operation of Thread %d: %d\n",x[0],var);
        printf("Operation Done by Thread %d...\n",x[0]);
    }
    else{
        printf("Entered in Thread :%d\n",x[0]);
        var-=x[1];
        printf("Value of var after the operation of Thread %d: %d\n",x[0],var);
        printf("Operation Done by Thread %d...\n",x[0]);
    }
}
```

Entered in Thread: 1
Value of var after operation of thread 1: 5
Operation Done by Thread 1

Entered in Thread: 2
Value of var after operation of thread 2: 2
Operation Done by Thread 2

Value of var after operation of threads: 2

$** \rightarrow$ value
 $* = \text{add.}$
 $\text{no stmt} \rightarrow \text{value}$



2. Find outputs of the following code. [Run this code in the PC multiple times and analyse the outputs]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
int t_id[]={1,2,3};
var=50;
void *t_func(int *v);
int main(){
    pthread_t t[3];
    for(int i=0;i<3;i++){
        pthread_create(&t[i],NULL,(void *)t_func,&t_id[i]);
        pthread_join(t[i],NULL);
    }
    printf("Final value of var: %d\n",var);
    return 0;
}
void *t_func(int *v){
    if(*v==0){
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var+=5;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
    else if(*v==1){
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var-=4;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
    else{
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var*=2;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
}
```

Entered in Thread 1
Thread 1 modified value 46
Thread 1 modified value 42
Thread 1 modified value 38
Modification done by Thread 1, value 38

Entered in Thread 2
Thread 2 modified value 76
Thread 2 modified value 152
Thread 2 modified value 304
Modification done by Thread 2, value 304

Entered in Thread 3
Thread 3 modified value 608
Thread 3 modified value 1216
Thread 3 modified value 2432
Modification done by Thread 3, value 2432

Final value of var: 2432

3. Find outputs of the following code. [Run this code in the PC multiple times and analyse the outputs every time]

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
int t_id[]={1,2,3};
var=50;
void *t_func(int *v);
int main(){
    pthread_t t[3];
    for(int i=0;i<3;i++){
        pthread_create(&t[i],NULL,(void *)t_func,&t_id[i]);
    }
    for(int i=0;i<3;i++){
        pthread_join(t[i],NULL);
    }
    printf("Final value of var: %d\n",var);
    return 0;
}

void *t_func(int *v){
    if(*v==0){
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var+=5;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
    else if(*v==1){
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var-=4;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
    else{
        printf("Entered in Thread %d...\n",*v);
        for(int i=0;i<3;i++){
            var*=2;
            printf("Thread %d modified value %d\n",*v,var);
        }
        printf("Modification done by Thread %d, value %d\n",*v,var);
    }
}
```

Handwritten annotations in red:

- 0, 1, 2 (next to the first for loop in main)
- 1, 2, 3 (next to the t_id array)
- Checkmarks (✓) next to the pthread_create and pthread_join calls.
- Red circles around &t_id[i] in pthread_create and pthread_join.