

```
//  
// Creazione di thread in ambiente Windows  
  
// Creazione di 2 thread per la ricerca di un valore fornito da tastiera  
// all'interno di un vettore caricato con numeri casuali.  
// Ogni thread ricerca il valore in metà vettore.  
// (viene utilizzata l'istruzione del linguaggio C printf provare a sostituirla con cout)  
//  
// Vengono richiamate le funzioni API di Windows per il multithreading:  
// CreateThread(...), WaitForSingleObject(...), GetCurrentThreadId(), ExitThread(...)  
  
//#include <iostream>  
#include <stdio.h>  
#include <stdlib.h>  
#include <windows.h>  
#include <time.h>  
//using namespace std;  
  
#define DIM 100000000  
  
int v[DIM]; // globale  
  
// struttura che definisce i parametri della funzione search  
struct PAR  
{  
    int first_index;  
    int last_index;  
    int search_value;  
};  
  
// prototipo  
unsigned long WINAPI search(void* );  
  
// ----- programma principale  
int main(void)  
{  
    int valore;  
    struct PAR par_1, par_2;  
    HANDLE thread_1, thread_2; // si predispongono due Handle per i thread che verranno creati  
  
    srand(time(NULL));  
    // generazione casuale dei valori degli elementi del vettore  
    for (int i=0; i<DIM; i++)  
    {  
        v[i] = rand()%100000;  
    }  
  
    // Richiesta valore da ricercare  
    printf("Inserire il numero da ricercare (0-99999): ");  
    scanf("%i", &valore);  
    //cout<<"Inserire il numero da ricercare (0-99999) ";  
    //cin>>valore;
```

```

// estremi della sezione di ricerca assegnata al primo thread
par_1.first_index = 0;
par_1.last_index = DIM/2-1;
par_1.search_value = valore;

// estremi della sezione di ricerca assegnata al secondo thread
par_2.first_index = DIM/2 ;
par_2.last_index = DIM-1;
par_2.search_value = valore;

// creazione dei thread e attesa della loro terminazione
thread_1 = CreateThread(NULL, 4096, &search, &par_1, 0, NULL);
thread_2 = CreateThread(NULL, 4096, &search, &par_2, 0, NULL);

WaitForSingleObject(thread_1, INFINITE);
WaitForSingleObject(thread_2, INFINITE);

system("pause");
}

// funzione che implementa il codice dei thread
unsigned long WINAPI search(void* arg)
{
int t_Id; // Identificatore del thread

PAR *tmp = (PAR*)(arg); // tmp puntatore all'oggetto di tipo PAR
PAR par = *tmp; // alla variabile par viene assegnata la struttura
// ricevuta come parametro della funzione

t_Id= GetCurrentThreadId(); // Restituisce l'identificatore del thread

unsigned long trovati=0;

// Ricerca del valore
for (int i=par.first_index; i<=par.last_index; i++)
if (v[i] == par.search_value)
{printf("T_ID %i -> indice =%i\r\n", t_Id,i);
//cout<<"T_ID "<<t_Id<<"\t"<<"-> indice= "<<i<<endl ;
trovati++;
}
//cout<<"\n"<<"-----> N. trovati "<<trovati<<endl;
printf("\n----->N.trovati %i \n", trovati);
ExitThread(0); //Terminazione del thread
}

```