

# Verifica di TPSIT, classe 4<sup>B</sup>ROB.

1. Scrivi in tutti i modi che conosci un ciclo do while che stampi tutti i valori contenuti nel vettore `int vet[10];`

1

```
int i = 0;
do{
    printf("%d ", vet[ i ]);
    i++;
}while(i < 10)
```

2

```
int i = 0;
do{
    printf("%d ", *(vet+i));
    i++;
}while(i < 10)
```

3

```
int* pi;
pi = vet;
do{
    printf("%d ", *pi);
    pi++;
}while(pi < vet+10)
```

4

```
int* pi;
int i = 0
pi = vet;
do{
    printf("%d ", *(pi+i));
    i++;
}while(i < 10)
```

2. Quali operazioni conosci che possono essere applicate ai puntatori? Fai un esempio dettagliato per ognuna di esse.

Le operazioni che possono essere applicate ai puntatori sono:

- l'operazione di dereferenziazione ovvero ricavare la variabile dato il suo indirizzo. Un esempio può essere:

```
int* pi;  
int a, b;  
pi = &a;  
b = *pi;
```

- l'operazione di assegnamento ovvero assegnare un determinato indirizzo ad un puntatore:

```
int* pi;  
int* r;  
int y;  
r = &y;  
pi = r;
```

- l'operazione di confronto ovvero verificare se due o più puntatori sono uguali, diversi, uno maggiore dell'altro o uno minore dell'altro:

```
int* pi;  
int* r;  
int y;  
r = &y;  
pi = r;  
if(pi == r){  
    printf("I due puntatori sono uguali");  
}
```

- somma e differenza tra numeri interi ad esempio:

```
int vet[10];  
int* pi;  
pi = vet;  
pi = pi+5;
```

- somma e differenza con altri puntatori ad esempio:

```
int vet[10];  
int* pi;  
int offset = 0;  
pi = vet;  
pi = pi+5;  
offset = pi - vet;
```

- l'operazione di ricavare un indirizzo di una variabile;  
int i = 5;  
int\* pi;  
**pi = &i;**

Le operazioni elencate sono quelle in grassetto

3. Scrivere un snippet di codice che dato il vettore

```
int vet[10] = {-3,1,4,-8,6,5,1,-10,0,1};
```

calcoli la media aritmetica dei valori in esso contenuto, utilizzando soltanto puntatori ed aritmetica dei puntatori.

```
int* pi;  
int somma = 0, media;  
for(pi = vet; pi < vet+10 ; pi++){  
  
    somma += *pi;  
}  
media = somma/10;
```

4. Che differenza c'è tra le due dichiarazioni:

- `float* a;`

- `float a[100];`

La differenza è che nel primo si dichiara un puntatore a float, ovvero una variabile che può contenere l'indirizzo di un'altra variabile di tipo float, mentre nel secondo un vettore ovvero un insieme di 100 variabili in cui in ognuna può essere contenuta un variabile di tipo float.

Da un punto di vista però della singola variabile a non c'è differenza perché entrambe sono dei puntatori a float, nella seconda però so che per tutto il codice a conterrà l'indirizzo della prima cella del vettore (`&a[0]`).

5. Dato il codice seguente, verificare che tutte le righe siano corrette e descrivere il significato di ciascuna riga :

```
int v[5] = {5,4,3,2,1}; //Inizializzo un  
vettore e gli assegno i valori 5 4 3 2 1
```

```
int *p; //Inizializzo un puntatore ad interi
```

```
p = v; //assegno al puntatore p l'indirizzo  
della variabile puntata da v ovvero la prima  
cella del vettore ( v[0])
```

```
p++; //incremento il puntatore p, adesso punta  
alla seconda cella del vettore (v[1])
```

```
printf("%d",*(p+4)); //qui c'è un errore perchè  
chiedo di stampare una cella che non fa parte  
del vettore, ovvero la sesta (v[5])
```

```
v++; //qui c'è un errore perché non possono  
incrementare il puntatore del vettore.
```

```
printf("%p",v); //qui stampo, sempre se  
escludiamo la riga precedente perché sbagliata,  
l'indirizzo della prima cella del mio array  
(&v[0]).
```

Data: 06/11/2020



**NOTA: utilizzare la terminologia ed il linguaggio specifico idoneo ad un testo tecnico in ambito informatico/smartrobot. E' richiesta chiarezza e correttezza di linguaggio.**