Data: 06/11/2020

Verifica di TPSIT, classe 4^BROB.

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

```
#define lunghezza_vettore 10

int i=0;

while(i<lunghezza_vettore){
printf("%d", vet[i]);
i++;
}

i=0;
while(i<lunghezza_vettore){
printf("%d", *(vet+i));
i++;
}

int *p;
p=vet;
while(p<vet+lunghezza_vettore){
printf("%d", *p);
p++;
}
```

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

dereferenziazione → s=*p;

aritmetiche → es s=*p+*c;

confronto → es if(*p<*c){}
assegnamento → *p=s;

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 v=0;
int *somma;
somma=&v;
int *media;
media=&v;
for(int i=0; i<10; i++){
*somma=*somma+*(vet+i);
}
*media=*somma/10;
```

4. Che differenza c'è tra le due dichiarazioni:
   - `float* a;`
   - `float a[100];`

la prima è una la dichiarazione di un puntatore di tipo char, quindi un puntatore a una variabile di tipo char.

mentre la seconda è la dichiarazione di un vettore di char, dove al suo interno si potrà salvare un char per cella;

l'unica differenza è che dentro l'array di char tu puoi salvare più dati rispetto ad un puntatore a char; altrimenti non esistono differenze tanto che

in C, il nome di un vettore è un puntatore inizializzato alla prima cella del vettore ovvero quella con indice zero, a cui si va a sommare un offset per accedere alle diverse celle.

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};   /*dichiarazione di un vettore di grandezza 5 a cui viene già assegnato dei valori*/

int *p;   /*dichiarazione di un puntatore di tipo intero*/

p = v;    /*salvataggio dell'indirizzo della prima cella dell'array dentro il puntatore p, e quindi viene inizializzato p*/

p++;   /*si incrementa p facendo ciò si accede alla seconda cella dell'array, ovvero quella con indice 0, ovvero v[1]*/

printf("%d",*(p+4));        /* facendo questa operazione si va a stampare il valore puntato da *(p+4) solo che facendo ciò avviene un errore di segmentazione, perché si accede ad un ad area di memoria non consentita, perché si va oltre alla dimensione dell'array, perché l'indice massimo a cui si può arrivare utilizzando un array è : grandezza_array -1 ; dato che si parte dall'indice 0*/

v++;   /*si incrementa v, accedendo così alla cella dell'array con indice 1*/

printf("%p",v);    /*e così si va a stampare, utilizzando il %p, l'indirizzo della cella di
```

```
    memoria a cui corrisponde in questo momento v
    */
```

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