
```
void inizializzazioneStack(){
        SP=-1;
}
```

Prototipo della funzione:

void inizializzazioneStack()

Scopo della Funzione:
Inizializza la variabile (Stack pointer) a -1

Descrizione parametri:

NULL

Parametri da restituire:

NULL

```
int stackPieno(){
        if(SP==dim-1)
            return 0;//Stack Pieno
        else return 1;
}
```

Prototipo della funzione:

int stackPieno()

Scopo della Funzione:
La funzione restituisce 0 se lo stack e pieno e 1 se lo stack non e pieno

Descrizione parametri:
NULL

Parametri da restituire:

0 → stack pieno
1 → stack non pieno

```
int stackVuoto(){
        if(SP==-1) return 0;//Stack Vuoto
        else return 1;
}
```

Prototipo della funzione:

int stackVuoto()

Scopo della Funzione:
La funzione restituisce 0 se lo stack e vuoto e 1 se lo stack non e vuoto

Descrizione parametri:
NULL

Parametri da restituire:

0 → stack vuoto
1 → stack non vuoto

```
void Push(int valoreInserito){
		if(stackPieno()!=dim) {
			SP++;
			Stack[SP]=valoreInserito;
			}
		else printf("Stack Pieno\n\n");
}
```

Prototipo della funzione:

Void Push(int valore inserito)

Scopo della Funzione:
La funzione inserisce un valore dentro allo stack

Descrizione parametri:
Int valore inserito → il valore che desidera essere inserito dentro allo stack

Parametri da restituire:

NULL

```
int Pop(){
	int dato;
	if (stackVuoto()!=0){
			dato=Stack[SP];
			SP--;
			return dato;
		}
	else {
		printf("Stack Vuoto\n\n");
		return -1;
		}
}
```

Prototipo della funzione:

Int Pop()

Scopo della Funzione:
La funzione elimina un valore dentro allo stack→preciso (First In Last OUT)

Descrizione parametri:
NULL

Parametri da restituire:
Ritorna il valore se il stack non e vuoto.
Ritorna -1 se lo stack e vuoto.

```
int ElementiInseriti(){
   int i=0;
   while (SP >= i)
   {
      cout<<" Posizione : "<<i
<<" Valore : "<< Stack[i]<<endl;
      i++;
   }
   return i;
}
```

Prototipo della funzione:

Int ElementiInseriti()

Scopo della Funzione:
La funzione restituisce la positione della variabile SP (Stack pointer)

Descrizione parametri:
NULL

Parametri da restituire:
Ritorna il valore della positione SP.

```
int ElementiNonInseriti(){
   if (SP == -1)
   return dim-1;
   else
   return (dim - SP -1);
}
```

Prototipo della funzione:

Int ElementiNonInseriti()

Scopo della Funzione:
La funzione restituisce Numero elementi rimasti da inserire

Descrizione parametri:
NULL

Parametri da restituire:
Ritorna il numero di valori rimasti da inserire.

```
int menu(){
   int scelta = 0;
   cout<<"\n1. Visualizza stack\n";
   cout<<"2. Riempire stack\n";
   cout<<"3. Push\n";
   cout<<"4. pop\n";
   cout<<"5. Elementi rimasti da inserire \n";
   cout<<"6. Elementi inseriti\n";
   cout<<"7. Exit\n";
   cout<<"";
   cin>>scelta;
   return scelta;
};
```

Prototipo della funzione:

Int menu()

Scopo della Funzione:
Visualizza il menu e prende la scelta dell utente. Poi la fa ritornare.

Descrizione parametri:
NULL

Parametri da restituire:
Ritorna il valore della scelta dell utente.

```
void RiempiStack()
{
   while(SP != dim-1)
   {
         SP++;
         tack[SP]=rand() % 101;
    }
}
```

Prototipo della funzione:

Void RiempiStack()

Scopo della Funzione:
La funzione riempie lo stack con valori random

Descrizione parametri:
NULL

Parametri da restituire:
NULL