



Liste

Informazioni funzioni delle Liste



```
list Crea_Lista(int Dimensione)
{
    list Head = NULL;
    list Appoggio = NULL;
    list aiuto=NULL;
    Head = Appoggio=new nodo;
    int i=0;

    while ( i < Dimensione)
    {
        Appoggio->next = new nodo;
        Appoggio=Appoggio->next;
        Appoggio->next=NULL;
        i++;
    }

    return Head;
};
```

Prototipo della Funzione:

- *list Crea_Lista(int Dimensione)*

Scopo della Funzione:

- *Creare la lista in base alla dimensione inserita.*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta la dimensione della lista*

Modalità di Passaggio dei Parametri:

- *Dimensione viene passata per valore*

Restituzione:

- *Viene restituito un puntatore Head con l'inizio della lista (Viene restituita la lista).*

```
void Visualizza_Lista(list Head){  
    int i=0;  
    if (Head->next == NULL)  
        cout<<"Lista Vuota\n";  
    else  
        while (Head->next != NULL)  
        {  
            cout<<i+1<<"- NODO >>  
            "<<Head->next<<" Valore :  
            "<<Head->dato<<"\n";  
            Head = Head->next;  
            i++;  
        }  
};
```

Prototipo della Funzione:

- *void Visualizza_Lista(list Head)*

Scopo della Funzione:

- *Visualizzare la lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *NULL.*

```
void Carica_Dati_Lista(list &Head)
{
    list Appoggio;
    Appoggio = Head;
    while (Appoggio != NULL)
    {
        Appoggio->dato = rand() %101;
        Appoggio=Appoggio->next;
    }
};
```

Prototipo della Funzione:

- *void Carica_Dati_Lista(list &Head)*

Scopo della Funzione:

- *Caricamento della lista con numeri random*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista passata per riferimento*

Restituzione:

- *NULL.*

```

void Cerca_Valore(list Head,int valore){
    int cercato = 0;
    int i=1;
    if(Head == NULL)
    {
        cout<<"Lista Vuota";
    }
    else
    {
        do
        {
            if(Head -> dato == valore)
            {
                cout<<"E stato trovato "<<" Valore : "<<Head-
>dato<<"\n";
                cercato++;
            }
            Head = Head->next;
        }while (Head != NULL);
        if (cercato == 0)
            cout<<"Non e stato trovato nessuno\n";
    }
};

```

Prototipo della Funzione:

- *void Cerca_Valore(list Head,int valore);*

Scopo della Funzione:

- *Cerca un valore dentro alla lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*
- *Il Secondo Parametro rappresenta il valore*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista passata per valore*
- *Il Secondo Parametro rappresenta il valore e viene passato per valore*

Restituzione:

- *NULL.*

```

int Conta_Pari_Lista(list Head)
{
    int conta = 0;
    if (Head == NULL)
    {
        cout<<"LISTA VUOTA\n";
        return -1;
    }
    else
    {
        while(Head->next!=NULL){
            if(Head->dato != 0){ // Valore diverso da 0
                if((Head->dato % 2) == 0){
                    conta++;
                }
            }
            Head = Head->next;
        };
        cout<<"PARI "<<conta<<"\n";
        return conta;
    }
};

```

Prototipo della Funzione:

- *int Conta_Pari_Lista(list Head)*

Scopo della Funzione:

- *Contare I numeri pari della lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *Il numero di pari.*

```

int Conta_Dispari_Lista(list Head)
{
    int conta = 0;
    if (Head == NULL)
    {
        cout<<"LISTA VUOTA\n";
        return -1;
    }
    else
    {
        while(Head->next != NULL){
            if(Head->dato != 0){
                if((Head->dato % 2) != 0)
                    conta++;
            }
            Head = Head->next;
        };
        cout<<"DISPARI "<<conta<<"\n";
        return conta;
    }
};

```

Prototipo della Funzione:

- *int Conta_Dispari_Lista(list Head)*

Scopo della Funzione:

- *Contare I numeri Dispari della lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *Il numero di Dispari.*

```

float Media_Pari_Lista(list Head)
{
    int conta = 0;
    float somma=0;
    if (Head == NULL)
    {
        cout<<"LISTA VUOTA\n";
        return -1;
    }
    else
    {
        while(Head->next!=NULL){
            if(Head->dato != 0){ // Valore diverso da 0
                if((Head->dato % 2) == 0){
                    somma = somma + Head->dato;
                    conta++;
                }
            }
            Head = Head->next;
        };
        return somma / conta ;
    }
};

```

Prototipo della Funzione:

- *float Media_Pari_Lista(list Head)*

Scopo della Funzione:

- *Fare la media dei pari e restituirla*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *La media dei Pari.*


```

float Media_Dispari_Lista(list Head)
{
    int conta = 0;
    float somma=0;
    if (Head == NULL)
    {
        cout<<"LISTA VUOTA\n";
        return -1;
    }
    else
    {
        while(Head->next!=NULL){
            if(Head->dato != 0){ // Valore diverso da 0
                if((Head->dato % 2) != 0){
                    somma = somma + Head->dato;
                    conta++;
                }
            }
            Head = Head->next;
        };
        return somma / conta ;
    }
};

```

Prototipo della Funzione:

- *float Media_Dispari_Lista(list Head)*

Scopo della Funzione:

- *Fare la media dei Dispari e restituirla*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *La media dei Dispari.*

```

list Crea_List_Pari(list Head)
{
    list Pari = NULL;
    list Appoggio = new nodo;
    Pari = Appoggio;
    if (Head == NULL)
    {
        return NULL;
    }
    else
    while (Head->next != NULL)
    {
        if (Head->dato % 2 == 0)
        {
            Appoggio->dato = Head->dato;
            Appoggio->next = new nodo;
            Appoggio = Appoggio->next;
        }
        Head = Head -> next;
    }
    return Pari;
};

```

Prototipo della Funzione:

- *list Crea_List_Pari(list Head)*

Scopo della Funzione:

- *Creare una lista pari e restituirla*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *La lista degli pari*

```

list Crea_List_Dispari(list Head)
{
    list Dispari = NULL;
    list Appoggio = new nodo;
    Dispari = Appoggio;
    if (Head == NULL)
    {
        return NULL;
    }
    else
    while (Head->next != NULL)
    {
        if (Head->dato % 2 != 0)
        {
            Appoggio->dato = Head->dato;
            Appoggio->next = new nodo;
            Appoggio = Appoggio->next;
        }
        Head = Head -> next;
    }
    return Dispari;
};

```

Prototipo della Funzione:

- *list Crea_List_Dispari(list Head)*

Scopo della Funzione:

- *Creare una lista dispari e restituirla*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *La lista degli dispari*

```
int Cerca_Valore_Massimo(list Head)
{
    int massimo = 0;
    if(Head==NULL)
        return -1;
    else
    {
        massimo = Head->dato;
        while(Head != NULL)
        {
            if (massimo < Head->dato)
            {
                massimo = Head->dato;
            }
            Head = Head->next;
        }
        return massimo;
    }
};
```

Prototipo della Funzione:

- *int Cerca_Valore_Massimo(list Head)*

Scopo della Funzione:

- *Cercare il valore massimo di una lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *Il valore massimo*

```
int Cerca_Valore_Minimo(list Head)
{
    int massimo = 0;
    if(Head==NULL)
        return -1;
    else
    {
        massimo = Head->dato;
        while(Head != NULL)
        {
            if (massimo > Head->dato)
            {
                massimo = Head->dato;
            }
            Head = Head->next;
        }
        return massimo;
    }
};
```

Prototipo della Funzione:

- *int Cerca_Valore_Minimo(list Head)*

Scopo della Funzione:

- *Cercare il valore minimo di una lista*

Descrizione dei Parametri:

Significato dei Parametri

- *Il Primo Parametro Rappresenta lista*

Modalità di Passaggio dei Parametri:

- *Il Primo Parametro Rappresenta lista pasata per valore*

Restituzione:

- *Il valore minimo*

```
int menu()
{
    int scelta = 0;
    cout<<"<-----Menu----->\n";
    cout<<" 1. Carica Lista ->RANDOM"<<endl;
    cout<<" 2. Visualizza Lista"<<endl;
    cout<<" 3. Cerca un valore"<<endl;
    cout<<" 4. Conta pari della lista"<<endl;
    cout<<" 5. Conta dispari della lista"<<endl;
    cout<<" 6. Media Pari"<<endl;
    cout<<" 7. Media Dispari"<<endl;
    cout<<" 8. Crea lista pari"<<endl;
    cout<<" 9. Crea lista dispari"<<endl;
    cout<<"10. Cerca Massimo"<<endl;
    cout<<"11. Cerca Minimo"<<endl;
    cout<<"12. EXIT"<<endl;
    cout<<"Inserisci Scelta : ";
    cin>>scelta;
    cout<<"\n";
    return scelta;
};
```

Prototipo della Funzione:

- *int menu()*

Scopo della Funzione:

- *Visualizzare il menu e restituire la scelta*

Descrizione dei Parametri:

Significato dei Parametri

- *NULL*

Modalità di Passaggio dei Parametri:

- *NULL*

Restituzione:

- *Scelta dell utente*