Verifica di TPSIT, classe 4^BROB.

1. Scrivere una funzione **ricorsiva** chiamata primo\_Dispari che data una Lista, che per valori ha numeri interi, restituisca il puntatore al primo elemento dispari nella lista (restituisce NULL se la lista è vuota o non contiene elementi pari).

struct Nodo primo\_Dispari( nodo\* primo){

int a;

nodo \*vettore=primo;

while (vettore -> next != NULL){

a = vettore -> valore % 2;

if( a == 1)

{

return vettore;

}

}

return NULL;}

1. Scrivere una funzione iterativa, oppure ricorsiva (a tua scelta), cancella, che ricevuta una Lista e un intero n, elimini (**senza deallocare**) i primi n elementi e ritorni il puntatore alla testa della lista modificata .

nodo \*cancella(int n, int b){

while(b<n)

{

head -> valore = NULL;

cancella(head -> next);

b++;

}

1. Data la struttura:

typedef struct nodo

{

int valore;

struct nodo\* successivo;

} Nodo;

e utilizzando la funzione (NON dovete implementarla):

Nodo\* push(Nodo\* lista, int x); //inserisce un elemento in fondo alla lista con il valore x e ritorna la head alla nuova lista

scrivere un programma che data una Lista di lunghezza arbitraria, crei una nuova lista che sia la copia esatta di Lista.

1. Quali vantaggi ha una lista rispetto ad un array?

i vantaggi delle liste in informatica rispetto agli array sono 2; le liste sono molto più veloci nell’inserimento e nella rimozione di elementi all’interno della lista perché ad esempio se bisogna inserire una elemento a metà dei una lista, basterà cambiare un puntatore e far puntare il nuovo elemento a quello successivo. Invece se devo inserire un elemento a metà di un array di tot caratteri è un procedimento molto più lungo perchè devo far scivolare verso destra un gran numero di elementi.