

```

// PROGRAMMA CHE ALLOCA DINAMICAMENTE UNA STRINGA E VISUALIZZA LA POSIZIONE IN MEMORIA
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char *s1;
    char buffer[80];
    int i;
    int x;
    int residuo;

    printf("inserire stringa \t ");
    scanf("%s", buffer);

    x=strlen(buffer);

    residuo=80-x;
    printf("\n \n \n il residuo e' %d\n", residuo);

    if(residuo>0){

        s1=(char *) malloc(x*sizeof(char));

        strcpy(s1, buffer);
    }
    else(exit(EXIT_FAILURE));

    for(i=0; s1[i]!='\0'; i++) printf("%d\t %d\t %c\n", i, &s1[i], s1[i]);

    free(s1);

    return 0;
}

```

//ELIMINAZIONE DI UN CARATTERE CON ALLOCAZIONE DINAMICA DELLA MEMORIA, MEDIANTE L'UTILIZZO DI UN BUFFER TEMPORANEO

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char *s1, ch;
    char stringa_temp[100];
    int x;
    int i, j, m;
    int residuo;

    printf("inserire la stringa: \t");
    scanf("%s", stringa_temp);

    x=strlen(stringa_temp);

    residuo=100-x;

    if(residuo>0){

        s1=(char *)malloc(x*sizeof(char));

        if(s1==NULL){
            printf("ALLOCAZIONE ERRATA \n");
            exit(EXIT_FAILURE);
        }

        strcpy(s1, stringa_temp);
    }
    else{
        printf("La stringa ha superato i 100 byte che ho messo a disposizione");
        exit(EXIT_FAILURE);
    }
}
```

```

printf("inserire il carattere da eliminare \t");
scanf(" %c", &ch);

for(i=0; s1[i]!='\0'; i++){
    if(s1[i]==ch){
        for(j=i; s1[j]!='\0'; j++) {
            s1[j]=s1[j+1];

        }
        i--;
    }
}
s1=(char *)realloc(s1, strlen(s1)+1);
printf("stringa risultante: ");
for(m=0; s1[m]!='\0'; m++) printf(" %c", s1[m]);
return 0;
}

```

// PROGRAMMA CHE INSERITA UNA STRINGA MINUSCOLA LA RISCRIVE TUTTA IN MAIUSCOLO

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char *s;
    char buffer[100];
    int i;
    printf("inserire la stringa: ");
    scanf("%s", buffer);

    s=(char *)malloc(sizeof(char)*strlen(buffer));
    strcpy(s, buffer);

    for(i=0; s[i]!='\0'; i++){

```

```

        if((s[i]>='a') &&(s[i]<='z')){
            s[i]=s[i]-32;
        }
    }
}

```

```

for(i=0; s[i]!='\0'; i++) printf("%c", s[i]);

```

```

    return 0;
}

```

// PROGRAMMA CHE TRASFORMA STRINGA MINUSCOLA IN MAIUSCOLA E VICEVERSA

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char *s1;
    char buffer[100];
    int i;
    int ok;

    printf("inserire stringa ");
    scanf("%s", buffer);

    s1=(char *)malloc(sizeof(char)*strlen(buffer));
    strcpy(s1, buffer);

    //LAVORIAMO SU S1 CHE E' STATO ALLOCATO DINAMICAMENTE

    for(i=0; s1[i]!='\0'; i++){
        if((s1[i]>='a') &&(s1[i]<='z')){
            s1[i]=s1[i]-32;
        }
    }
}

```

```

        else {
            s1[i]=s1[i]+32;
        }
    }
}

```

```

for(i=0; s1[i]!='\0'; i++)printf("%c", s1[i]);

```

```

return 0;
}

```

```

// PROGRAMMA CHE CONTA LETTERE NUMERI E CARATTERI SPECIALI

```

```

#include <stdio.h>

```

```

#define MAX_SIZE 100 // Maximum string size

```

```

int main()

```

```

{

```

```

    char str[MAX_SIZE];

```

```

    int alphabets, digits, others, i;

```

```

    alphabets = digits = others = i;

```

```

    printf(" inserire stringa : ");

```

```

    scanf("%s", str);

```

```

    for(i=0; str[i]!='\0'; i++){

```

```

        if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))

```

```

        {

```

```

            alphabets++;

```

```

        }
    }
}

```

```

        else if(str[i]>='0' && str[i]<='9')
        {
            di gi ts++;
        }
        else
        {
            others++;
        }

        i++;
    }

    printf("Al phabets = %d\n", al phabets);
    printf("Di gi ts = %d\n", di gi ts);
    printf("Special characters = %d", others);

    return 0;
}

```

//PROGRAMMA CHE CONTA LE OCCORRENZE DELLE VOCALI DATE IN INPUT

```
#include <stdio.h>
```

```
#define MAX_SIZE 100 // Maximum string size
```

```
int main()
```

```
{
```

```
    char str[MAX_SIZE];
```

```
    int alphabets, digits, others, i;
```

```
    int vocali=0;
```

```
    printf(" inserire stringa : ");
```

```
    scanf("%s", str);
```

```
    for(i=0; str[i]!='\0'; i++){
```

```
        if((str[i]=='a') || (str[i]=='e') || (str[i]=='i' ) ||
(str[i]=='o') || (str[i]=='u'))
```

```

        {
            vocal i++;
        }
    }

    printf("vocal i   = %d\n", vocal i);

    return 0;
}

```

// PROGRAMMA CHE VISUALIZZA LA STRINGA AL CONTRARIO

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100
int main()
{
    char str[MAX], reverse[MAX];
    int i;
    int lung;
    int x;

    printf("inserire stringa : ");
    scanf("%s", str);

    reverse[0]='\0';

    lung=strlen(str);

    for(i=0; str[i]!='\0'; i++){

        reverse[i] = str[lung-1];
        lung--;

    }
    reverse[i]='\0';
}

```

```
for(x=0; reverse[x]!='\0'; x++) printf("%c", reverse[x]);
```

```
return 0;
```

```
}
```

```
// CONTROLLO DELLA PAROLA PALINDROMA
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define MAX 100
```

```
int main()
```

```
{
```

```
    char str[MAX], reverse[MAX];
```

```
    char buffer[100];
```

```
    int i;
```

```
    int lung;
```

```
    printf("inserire stringa : ");
```

```
    scanf("%s", str);
```

```
    strcpy(buffer, str);
```

```
    reverse[0]='\0';
```

```
    lung=strlen(str);
```

```
    for(i=0; str[i]!='\0'; i++){
```

```
        reverse[i] = str[lung-1];
```

```
        lung--;
```



```

    }
    reverse[i]='\0';
printf("\n la stringa al contrario e' ");
    for(i=0; reverse[i]!='\0'; i++) printf("%c", reverse[i]);

    for(i=0; buffer[i]!='\0'; i++){
        if(buffer[i]==reverse[i]){
            printf(" \n parola palindroma");
        }
        else printf("\n parola non palindroma ");

break;

    }

    return 0;
}

// PROGRAMMA CHE DIVIDE UNA STRINGA IN TOKEN USANDO LA MATRICE

#include <stdio.h>
#include <string.h>

#define DELIM " "
#define PAROLEMAX 80
#define LUNGHEZZAMAX 20

int main(void)
{
    char words[PAROLEMAX][LUNGHEZZAMAX];
    char buff[100];
    int numtok = 0;
    int i;
    char *sep;

    printf("inserire stringa : ");

```

```
gets(buff);
```

```
sep = strtok(buff, DELIM);
```

```
while (sep != NULL)
{
    strcpy(words[numtok++], sep);
    sep = strtok(NULL, DELIM);
}
```

```
for (i = 0; i < numtok; i++) {
```

```
    printf("\n%d %s\t", i, words[i]);
```

```
}
```

```
}
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char linea[100];
```

```
    char *ptr;
```

```
    char c;
```

```
    int i=0;
```

```
    ptr=(char *) malloc(sizeof(char));
```

```
    ptr[0]='\0';
```

```
    if(ptr!=NULL){
```

```
        printf("--->allocazione avvenuta in modo corretto <-----");
```

```

    }
    else exit(EXIT_FAILURE);

    printf("\n \n \n inserisci una linea che termina con invio su riga
    successiva \n");
    while((c=getchar())!=EOF){
        if(c!='\n'){
            linea[i++]=c;
        }
        else{
            linea[i]=' ';
            linea[i+1]='\0';

            ptr=(char *)realloc(ptr, strlen(ptr)+strlen(linea)+1);
            strcat(ptr, linea);
        }
    }

    if(ptr!=NULL){
        printf("--->allocazione avvenuta in modo corretto <-----");
        printf("\n stringa risultante= %s", ptr);
    }
    else exit(EXIT_FAILURE);

    return 0;
}

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()

```

```

{
    char linea[100];
    char accumul o[100];
    int i=0, a;
    char c;
    linea[0]='\0';
    accumul o[0]='\0';

    printf("inserire la linea ");
    while((c=getchar())!=EOF){
        if(c!='\n'){
            linea[i++]=c;
        }
        linea[i]=' ';
        linea[i+1]='\0';
    }

    for(a=0; linea[a]!='\0'; a++) printf("%d %c\n", a, linea[a]);

    return 0;
}

```

FILE A TOKEN

```

/*
    Lettura di un file riga per riga.
*/

#include<stdlib.h>
#include<stdio.h>
#include<string.h>

int main() {
    FILE *fd;
    char matrix[100][200];

```

```

char buf[200];
char *res;
char token;
char *s="\n";
int i, n=0;


/* apre il file */
fd=fopen("righe.txt", "r");
if( fd==NULL ) {
    perror("Errore in apertura del file");
    exit(1);
}


/* legge e stampa ogni riga */
while(fgets(buf, 200, fd)!=NULL){
    res=(char *)malloc(strlen(buf)*sizeof(char));
    strcpy(res, buf);
token=strtok(res, s);

while(token!=NULL){
    strcpy(matrix[n++], res);
    token=strtok(NULL, s);
}

}

for(i=0; i<n; i++)printf("\n%d %s\t", i, matrix[i]);

/* chiude il file */
fclose(fd);

return 0;
}

/*

```

Lettura di un file riga per riga.

*/

```
#include<stdlib.h>
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
int main() {
```

```
    FILE *fd;
```

```
    char matrix[100][200];
```

```
    char buf[200];
```

```
    char *res;
```

```
    char *s=" ";
```

```
    int i, n=0;
```

```
        /* apre il file */
```

```
    fd=fopen("righe.txt", "r");
```

```
    if( fd==NULL ) {
```

```
        perror("Errore in apertura del file");
```

```
        exit(1);
```

```
    }
```

```
        /* legge e stampa ogni riga */
```

```
    while(fgets(buf, 200, fd)!=NULL) {
```

```
        res=(char *)malloc(strlen(buf)*sizeof(char));
```

```
        strcpy(res, buf);
```

```
        for(i=0; res[i]!='\0'; i++)printf("%c", res[i]);
```

```
    }
```

```
    return 0;
```

```
    }
```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char v1[20];

    int x;
    int i;

    printf("Inserisci la stringa \n");
    gets(v1);
    x=strlen(v1);

    for(i=0; i<x; i++){
        printf("elemento %d\t carattere %c\n", i, v1[i]);

        if((v1[20]>='a') || (v1[20]<='z')){
            v1[0]=' ';
        }
    }
    printf("\n \n \n");
    for(i=0; i<x; i++){

        printf("elemento %d\t carattere %c\n", i, v1[i]);

    }

    return 0;
}

```

```

#include <stdio.h>

```

```

#include <stdlib.h>
# include <string.h>
int main()
{
    char v1[20];

    int x;
    int i;

    printf("Inserisci la stringa \n");
    gets(v1);
    x=strlen(v1);

    for(i=0; i<x; i++){
        printf("elemento %d\t carattere %c\n", i, v1[i]);

        if((v1[20]>='a') || (v1[20]<='z')){
            v1[x-1]=' ';
        }
    }
    printf("\n \n \n");
    for(i=0; i<x; i++){

        printf("elemento %d\t carattere %c\n", i, v1[i]);

    }

    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>
# include <string.h>
int main()

```



```

{
    char v1[20];

    int x;
    int i;

    printf("Inserisci la stringa \n");
    scanf("%s", v1);
    x=strlen(v1);

    for(i=0; i<x; i++){
        printf("elemento %d\t carattere %c\n", i, v1[i]);

        if(x%2==0){
            if((v1[20]>='a') || (v1[20]<='z')){
                v1[(x-1)/2]=' ';
            }

            } // fine if

        } // fine for

```

```

        return 0;
    }

```

```

-----
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

#define N 300
int main()
{

```

```

char v1[N];
char v2[N];
int n;
int i;
int j;
int r;
int rimanenza;
printf("inserire il numero di stinghe \n");
scanf("%d", &n);
v2[0]='\0';
for(i=0; i<n; i++){
    printf("\n stringa %d\t", i+1);
    scanf("%s", v1);

    r=strlen(v2);
    printf("v2=%d ", r);
    j=strlen(v1);
    printf("v1=%d ", j);

    rimanenza=N- strlen(v1)- strlen(v2);
    printf("La rimanenza e' di %d      ", rimanenza);

    if(rimanenza>0){
strcat(strcat(v2, " "), v1);

    }

    else{
        exit(0);
    }
    printf("in V2 tramite la funzione strcat avro' %s", v2);

}
return 0;

```

```
}
```

```
-----  
-
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
void elimina(char*, char );
```

```
int main()
```

```
{
```

```
    char linea[100], c;
```

```
    int i, n=0;
```

```
    printf("inserire linea \n");
```

```
    while(((c=getchar())!='\n')&&(n<99)) linea[n++]=c;
```

```
    linea[n]='\0';
```

```
    printf("inserire il carattere da eliminare \n");
```

```
    scanf(" %c", &c);
```

```
n=elimina(linea, c);
```

```
printf("il numero delle occorrenze e' %d", n);
```

```
printf("linea risultante : ");
```

```
for(i=0; linea[i]!='\0'; i++) printf("%c", linea[i]);
```

```
    return 0;
```

```
}
```

```

void elimina(char*s, char ch ){
int i,j;
int m=0;

for(i=0; s[i]!='\0'; i++){
    if(s[i]==ch){
        for(j=i; s[j]!='\0'; j++){
            s[j-1]=s[j];
        }
        i--;
        m++;
    }
}

}

}

```

```

-----

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char stringa[100];
    char ch;
    int i,j;

    printf("inserire stringa \n :");
    scanf("%s", stringa);

    fflush(stdin);

```

```
printf("inserire il carattere da cercare \n");  
scanf("%c", &ch);
```

```
for(i=0; stringa[i]!='\0'; i++){  
    if(stringa[i]==ch) {  
        for(j=i; stringa[j]!='\0'; j++) {  
            stringa[j-1]=stringa[j];  
        }  
    }  
}
```

```
for(i=0; stringa[i]!='\0'; i++) {  
    printf("%c", stringa[i]);  
}  
return 0;  
}
```