Table of Contents

Write a Program to Convert Lower to Upper &Upper to Lower Cases of a given file.

```
#include<stdio.h>


main(int argc,char**argv)

{

        FILE *fp;

        char ch;


        if(argc!=2)

        {

        printf("Usage : ./a.out fiename\n");

        return;

        }


        fp=fopen(argv[1],"r+");


        if(fp==0)

        {

        printf("File does not exist\n");

        return;

        }


        while((ch=fgetc(fp)) != EOF)
        {
        if((ch >= 97 && ch <=122) || (ch>=65 && ch <= 90))
        ch=ch^32;
        fseek(fp,-1,SEEK_CUR);
        fputc(ch,fp);
        }
}
```

Write a Program to remove a Specific line from the given text file.

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

main(int argc,char**argv)
{
        FILE *fp;
        char ch,**cp;
        int c,k,i,m,c1,max;

        if(argc != 3)
        {
        printf("Usage : ./a.out filename\n");
        return;
        }

        fp=fopen(argv[1],"r+");
        if(fp==0)
        {
        printf("File is not presentn\n");
        return;
        }
        max = c1 = c = k = 0;
        while((ch=fgetc(fp)) != EOF)
```

```
{
c++;
c1++;
if(ch==10)
{
        if(c1>max)
        max=c1;
        k++;
        c1=0;
}
}
rewind(fp);
m=atoi(argv[2]);
m=m-1;
printf("SIZE : %d Lines : %d Max : %d\n",c,k,max-1);

cp = malloc(sizeof(char*) * k);

for(i=0;i<k;i++)
{
cp[i]=malloc(max+1);
}

for(i=0;i<k;i++)
```

```
        {
        fgets(cp[i],max+1,fp);
        }
        fclose(fp);

        fp=fopen(argv[1],"w");
        for(i=0;i<k;i++)
        {
        if(m!=i)
        fputs(cp[i],fp);
        }

}
```

Write a Program to replace a Specified line in a given file.

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

main(int argc,char**argv)
{
        FILE *fp;
        int i,c,k,c1,max,m;
        char ch,**cp,*p;

        if(argc!=3)
        {
        printf("Usage : ./a.out filename line\n");
        return;
        }

        fp=fopen(argv[1],"r");
        if(fp==0)
        {
        printf("File doesn`t exist\n");
        return;
        }
        m=atoi(argv[2]);
        m=m-1;
```

```
max=c1=c=k=0;
while((ch=fgetc(fp)) != EOF)
{
c++;
c1++;
if(ch==10)
{
        if(c1>max)
        max=c1;
        k++;
        c1=0;
}
}
printf("SIZE : %d Lines : %d Max : %d\n",c,k,max-1);
rewind(fp);
cp=malloc(sizeof(char*)*k);
for(i=0;i<k;i++)
{
cp[i]=malloc(max+1);
}

for(i=0;i<k;i++)
{
fgets(cp[i],max+1,fp);
```

```
	}

	p = malloc(c);

	printf("Enter String to Replace:\n");
	gets(p);

	fclose(fp);

	fp = fopen(argv[1],"w");

	for(i=0;i<k;i++)
	{
	if(i==m)
	fprintf(fp,"%s\n",p);
	else
	fputs(cp[i],fp);
	}

}
```

Write a Program to Capitalize First Letter of every Word in a file.

```
#include<stdio.h>

#include<stdlib.h>


main(int argc,char**argv)

{

        FILE *fp;

        int m,c;

        char ch,*cp;


        if(argc != 2)

        {

        printf("Usage : ./aout filename\n");

        return;

        }


        fp = fopen(argv[1],"r+");


        if(fp==0)

        {

        printf("File is not present\n");

        return;

        }
        c=0;

        while((ch=fgetc(fp)) != EOF)
        {
        c++;
        }
        rewind(fp);
        cp = malloc(c);

        while((fscanf(fp,"%s",cp)) != EOF)
        {
        for(m=0;cp[m];m++);
        cp[0] = cp[0] &~(32);
        fseek(fp,-m,SEEK_CUR);
        fputs(cp,fp);
        }

}
```

Write a Program to merges the lines from 2 files & store the result into another file.

```
#include<stdio.h>

#include<stdlib.h>


main(int argc,char**argv)

{

        FILE *fp1,*fp2,*fp;

        int i,j,k,l,m,n,c1,k1,c2,k2;

        char **cp1,**cp2,ch1,ch2;


        if(argc != 4)

        {

        printf("Usage : ./a.out file1 file2 destfile\n");

        return;

        }

        fp1 = fopen(argv[1],"r");

        fp2 = fopen(argv[2],"r");


        if(fp1==0 || fp2==0)

        {

        printf("File is not present\n");

        return;

        }
```

```
c1=c2=k1=k2=0;
while((ch1=fgetc(fp1)) != EOF)
{
c1++;
if(ch1==10)
k1++;
}
rewind(fp1);
while((ch2=fgetc(fp2)) != EOF)
{
c2++;
if(ch2==10)
k2++;
}
rewind(fp2);

cp1 = malloc(sizeof(char*) * k1);
for(i=0;i<k1;i++)
{
cp1[i]=malloc(c1);
}
cp2 = malloc(sizeof(char*) * k2);
for(i=0;i<k2;i++)
```

```
{
cp2[i]=malloc(c2);
}

for(i=0;i<k1;i++)
{
fgets(cp1[i],c1,fp1);
}
for(i=0;i<k2;i++)
{
fgets(cp2[i],c2,fp2);
}

fp = fopen(argv[3],"w");

for(i=0,j=0;i<k1 && j<k2;i++,j++)
{
fputs(cp1[i],fp);
fputs(cp2[j],fp);
}

if(i<k1)
{
for(;i<k1;i++)
```

```
        fputs(cp1[i],fp);
        }
        else if(j<k2)
        {
        for(;j<k2;j++)
        fputs(cp2[j],fp);
        }
}
```

Write a Program to replace the word with the reverse of that word in a given file.

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

main(int argc,char**argv)
{
        FILE *fp;
        char *cp,ch;
        int i,j,l;
        if(argc != 2)
        {
        printf("Usage : ./a.out filename\n");
        return;
        }

        fp = fopen(argv[1],"r+");

        if(fp == 0)
        {
        printf("File is not present\n");
        return;
        }
        fseek(fp,0,SEEK_END);
```

```
	l=ftell(fp);
	printf("Size : %d\n",l);

	cp = malloc(l);

	rewind(fp);

	while((fscanf(fp,"%s",cp)) != EOF)
	{
	for(i=0;cp[i];i++);
	fseek(fp,-i,SEEK_CUR);
	for(j=0,i=i-1;j<i;j++,i--)
	{
		ch = cp[j];
		cp[j] = cp[i];
		cp[i] = ch;
	}
	fputs(cp,fp);
	}
}
```

Write a Program to copy the one file into multiple destination files.

(which are provided during the load time.)

```
#include<stdio.h>


main(int argc,char**argv)

{

        FILE *fp,*fp1;

        char ch,i;


        if(argc < 3)

        {

        printf("Usage : ./a.out sfilename dfilename\n");

        return;

        }

        fp = fopen(argv[1],"r");


        for(i=2;i<=argc-1;i++)

        {

        fp1=fopen(argv[i],"w");


        while((ch=fgetc(fp)) != EOF)

        {

                fputc(ch,fp1);

        }
```

```
    rewind(fp);
    }
}
```

Write a Program to replace the particular word with another word in a given file.

```
#include<stdio.h>

#include<stdlib.h>


main(int argc,char**argv)

{

        FILE *fp;

        char l1,l2,l3,c,ch,*p,i,j,k;


        if(argc != 4)

        {

        printf("Usage: ./a.out filename sword dword\n");

        return;

        }


        fp = fopen(argv[1],"r");


        if(fp == 0)

        {

        printf("File is not present\n");

        return;

        }
```

```
for(i=0;argv[2][i];i++,l2=i);
for(i=0;argv[3][i];i++,l3=i);

c=0;

while((ch = fgetc(fp)) != EOF)
{
c++;
}
rewind(fp);

l1=c;

printf("l1 : %d\nl2 : %d\nl3 : %d\n",l1,l2,l3);

p = malloc(c+1);

i=0;

while((ch = fgetc(fp)) != EOF)
{
p[i++] = ch;
}
p[i]='\0';
```

```
for(i=0;p[i];i++)
{
if(p[i] == argv[2][0])
{
        for(j=1;argv[2][j];j++)
        if(argv[2][j] != p[i+j])
                break;

        if(argv[2][j] == '\0')
        {
                l1 = l1+l3-l2;

                for(c=l2;c>0;c--)
                        for(k=i;p[k];k++)
                                p[k]=p[k+1];

                for(c=l3;c>0;c--)
                        for(k=l1;k>=i;k--)
                                p[k+1]=p[k];

                for(k=0;k<l3;k++)
                        p[i+k]=argv[3][k];
        }
        }
        }

        fp = fopen(argv[1],"w");

        for(i=0;p[i];i++)
        {
        fputc(p[i],fp);
        }
}
```

Write a Program to reverse the Contents of a given file.

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

main(int argc,char**argv)
{
        FILE *fp;
        char j,c,k,i,**cp,c1,max,*p,m,ch;

        if(argc != 2)
        {
        printf("Usage : ./a.out filename\n");
        return;
        }

        fp=fopen(argv[1],"r");

        if(fp==0)
        {
        printf("File is not present\n");
        return;
        }
        c1 = max = c = k = 0;
        while((ch = fgetc(fp)) != EOF)
```

```
{
c1++;
c++;
if(ch==10)
{
	k++;
	if(c1>max)
	max=c1;
	c1=0;
}
}
rewind(fp);
cp = malloc(sizeof(char*)*k);

for(i=0;i<k;i++)
{
cp[i] = malloc(max+1);
}
for(i=0;i<k;i++)
{
fgets(cp[i],max+1,fp);
}
```

```
        for(i=0,m=k-1;i<m;i++,m--)
        {
        p=cp[i];
        cp[i]=cp[m];
        cp[m]=p;
        }
        fclose(fp);
        fp = fopen(argv[1],"w");
        for(i=0;i<k;i++)
        {
        fputs(cp[i],fp);
        }

}
```

Write a Program to implement sort command.

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>

main(int argc,char**argv)

{

        FILE *fp,*fp2;

        char ch,**cp,c1,c,k,max,i,j,*p;


        if(argc != 3)

        {

        printf("Usage : ./a.out srcfilename destfilename\n");

        return;

        }


        fp = fopen(argv[1],"r");

        if(fp==0)

        {

        printf("File doesn`t exist\n");

        return;

        }


        c1 = c = k = max = 0;

        while((ch =fgetc(fp)) != EOF)
```

```
{
c++;
c1++;
if(ch==10)
{
        if(c1>max)
        max=c1;
        c1 = 0;
        k++;
}
}
rewind(fp);
cp = malloc(sizeof(char *) * k);
for(i=0;i<k;i++)
{
cp[i]=malloc(max+1);
}
for(i=0;i<k;i++)
{
fgets(cp[i],max+1,fp);
}

for(i=0;i<k;i++)
{
        for(j=i+1;j<k;j++)
        {
                if(strlen(cp[i]) > strlen(cp[j]))
                {
                        p = cp[i];
                        cp[i]=cp[j];
                        cp[j]=p;
                }
        }
        }
        fp2=fopen(argv[2],"w");
        for(i=0;i<k;i++)
        {
        fputs(cp[i],fp2);
        }
}
```

Write a Program to implement grep command.

```
#include<stdio.h>

#include<string.h>


main(int argc,char**argv)

{

        FILE *fp;

        char a[100],*cp;


        if(argc != 3)

        {

        printf("Usage : ./a.out filename word \n");

        return;

        }


        fp = fopen(argv[1],"r");


        while(fgets(a,100,fp))

        {

        if(strstr(a,argv[2]))

        printf("%s",a);

        }

}
```

Write a Program to implement wc command.

```
#include<stdio.h>
main(int argc,char**argv)
{
        FILE *fp;
        char ch;
        int c,k,m;

        if(argc != 2)
        {
        printf("Usage : ./a.out filename\n");
        return;
        }

        fp=fopen(argv[1],"r");
        if(fp==0)
        {
        printf("file is not present\n");
        return;
        }
        c=k=m=0;
        while((ch=fgetc(fp)) != EOF)
        {
        c++;
        if(ch==32)
        m++;
        if(ch==10)
        {
                m++;
                k++;
        }
        }
        printf("%d %d %d %s\n",k,m,c,argv[1]);
}
```