

Датотеки

1. Да се напише програма која за дадена текстуална датотека чие име се внесува преку командна линија, ќе го одреди односот на самогласките и согласките.

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

int Samoglaska(char k) {
    switch (k) {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
        case 'A':
        case 'E':
        case 'I':
        case 'O':
        case 'U': return 1 ;
        default: return 0;
    }
}

int main(int argc,char *argv[]){
    char c;
    int brsogl=0, brsamogl=0;
    FILE *dat;
    if((dat=fopen(argv[1],"r"))==NULL){
        fprintf(stderr,"Ne mozam da ja otvoram datotekata %s",argv[1]);
        return -1;
    }
    while((c=fgetc(dat))!=EOF) /* citaj znak po znak se do krajot */
    {
        if (isalpha(c)){
            if (Samoglaska(c)) brsamogl++;
            else brsogl++;
        }
    }
    printf("Odnosot samoglaski/soglaski = ");
    printf("%d/%d=%5.2f\n",brsamogl, brsogl, (float)brsamogl/brsogl);
    return 0;
}
```

2. Да се напише програма која како аргумент од командна линија прима име на влезна датотека во која се сместени броеви на мобилни телефони, по еден број во ред. Програмата треба да ги запише броевите на мобилните телефони во излезна датотека, чие име исто така се задава како аргумент од командна линија. Дополнително после секој број, треба да се запишат цифрите од истиот, но со зборови, одделени со тире (-). Да се провери дали правилно се користи програмата, ако не, да се даде упатство за користење.

Пример за валидни аргументи внесени од командна линија:: ./a.out p1.txt p2.txt

Пример:

Влезна датотека:

070123456

072233456

Излезна датотека:

070123456-nula-sedum-nula-eden-dva-tri-cetiri-pet-sest

072233456-nula-sedum-dva-dva-tri-tri-cetiri-pet-sest

```

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

int main(int argc, char *argv[]){
    int i=0, j;
    char c;
    char broj[100];

    FILE *in_dat;
    FILE *out_dat;
    if(argc!=3){
        fprintf(stderr,"Upatstvo:\n %s ", argv[0]);
        fprintf(stderr, "<vlezna_datoteka> <zlezna_datoteka>\n");
        return -1;
    }
    if((in_dat=fopen(argv[1], "r"))==NULL){
        fprintf(stderr, "Ne mozam da ja otvoram datotekata ");
        fprintf(stderr, "%s za citanje\n", argv[1]);
        return -1;
    }
    if((out_dat=fopen(argv[2], "w"))==NULL){
        fprintf(stderr, "Ne mozam da ja otvoram datotekata ");
        fprintf(stderr, "%s za zapisuvanje\n", argv[2]);
        return -1;
    }

    while((c=fgetc(in_dat))!=EOF) {
        if((c>='0')&&(c<='9')){
            // citam cifra od mobilniot broj
            fputc(c,out_dat);
            broj[i++]=c;
        }
        else if(c=='\n'){// stignav do kraj na redot
            for(j=0;j<i;j++){
                switch(broj[j]){
                    case '0':fputs("-nula",out_dat);break;
                    case '1':fputs("-eden",out_dat);break;
                    case '2':fputs("-dva",out_dat);break;
                    case '3':fputs("-tri",out_dat);break;
                    case '4':fputs("-cetiri",out_dat);break;
                    case '5':fputs("-pet",out_dat);break;
                    case '6':fputs("-sest",out_dat);break;
                    case '7':fputs("-sedum",out_dat);break;
                    case '8':fputs("-osum",out_dat);break;
                    case '9':fputs("-devet",out_dat);break;
                    default: break;
                } //switch
            } //for(j)
            fputc(c,out_dat);
            printf("\n");
            i=0;
        } //else if
    } //while(c=fgetc...)
    fclose(in_dat);
    fclose(out_dat);
    return 0;
}

```

3. Да се напише програма која за дадена текстуална датотека, чие име се внесува како аргумент од командна линија, ќе ги најде сите редови кои имаат повеќе од 10 мали букви и во втора датотека (чие име исто така се внесува како аргумент од командна линија) ќе ги

испечати редните броеви на таквите редови. На крај во втората датотека да се испечати и вкупниот број на редови кои имаат повеќе од 10 мали букви.

Програмата треба да провери дали корисникот вnel влезна и излезна датотека (доколку нема внесено појавува порака за грешка).

Пример:

dat1.txt
Prva Linija

dat2.txt
3
Vкупно takvi redovi:1

Vtora linija so PoveKje bukvi

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main(int argc,char *argv[]){
    int red,mali,vkupno;
    char c;
    FILE *in_dat, *out_dat;

    if(argc!=3){
        fprintf(stderr,"Upatstvo:\n %s ",argv[0]);
        fprintf(stderr,"<vlezna_datoteka> <izlezna_datoteka>\n");
        return -1;
    }
    if((in_dat=fopen(argv[1],"r"))==NULL){
        fprintf(stderr,"Ne mozam da ja otvoram datotekata ");
        fprintf(stderr,"%s za citanje\n",argv[1]);
        return -1;
    }
    if((out_dat=fopen(argv[2],"w"))==NULL){
        fprintf(stderr,"Ne mozam da ja otvoram datotekata ");
        fprintf(stderr,"%s za zapisuvanje\n",argv[2]);
        return -1;
    }

    mali=vkupno=0; red=1;
    while((c=fgetc(in_dat))!=EOF){
        if(isalpha(c)&&islower(c)){
            mali++;
        }
        if(c=='\n'){
            if(mali>=10){
                fprintf(out_dat,"%d\n",red);
                vkupno++;
            }
            //if(mali)
            red++;
            mali=0;
        }
        //if(nov red)
    }
    //while(c=fgetc...)
    fprintf(out_dat,"Vкупно takvi redovi:%d\n",vkupno);
    fclose(out_dat);
    fclose(in_dat);
    return 0;
}
```

4. Да се напише програма која ќе ги прочита елементите од една матрица сместени во текстуалната датотека "matrica1.txt". На почетокот на датотеката се запишани бројот на редици и колони на матрицата. Секој елемент е реален број запишан во посебен

ред во датотеката. Потоа матрицата треба да се транспонира и да се запише во нова датотека "matrica2.txt" на истиот начин.

```
#include <stdio.h>
#define MAX 10
int main(){
    int i,j,m,n;
    float a[MAX][MAX], b[MAX][MAX];
    FILE *vlezna, *izlezna;

    if((vlezna=fopen("matrical.txt", "r"))==NULL){
        printf("Ne mozam da ja otvoram datotekata matrical.txt");
        return(-1);
    }

    if (!feof(vlezna)) fscanf(vlezna, "%d", &m);
    if (!feof(vlezna)) fscanf(vlezna, "%d", &n);
    if ((m>MAX) || (n> MAX)){
        printf("Mnogu golema matrica!");
        return(-1);
    }

    for (i=0; (i<m) && !feof(vlezna) ;i++)
        for (j=0; (j<n) && !feof(vlezna) ; j++)
            fscanf(vlezna, "%f", &a[i][j]);
    fclose(vlezna);

    if ((i!=m) || (j!=n)){
        printf("Nema dovolno podatoci vo datotekata!");
        return(-1);
    }

    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            b[j][i]=a[i][j];

    if((izlezna=fopen("matrica2.txt", "w"))==NULL){
        printf("Ne mozam da ja otvoram datotekata matrica2.txt");
        return(-1);
    }

    fprintf(izlezna, "%d\n", n); /* obratno */
    fprintf(izlezna, "%d\n", m);

    for (i=0;i<n;i++)
        for (j=0;j<m;j++)
            fprintf(izlezna, "%7.2f\n", b[i][j]);

    fclose(izlezna);

    return 0;
}
```

5. Да се напише програма која како параметар ќе го прима името на некоја датотека и ќе го отпечати бројот на редови во кои има по најмалку десет зборови. Зборовите се составени само од букви. Во секој ред има најмногу 80 знака.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 80
```

```

int main(int argc, char *argv[]){
    char s[MAX], *c;
    int zbor, red=0, vozbor;
    FILE *dat;

    if(argc!=2){
        printf("Upotreba: a.out ime_na_datotekata");
        return -1;
    }
    if((dat=fopen(argv[1], "r"))==NULL){
        printf("Ne mozam da ja otvoram datotekata %s\n", argv[1]);
        return -1;
    }

    while((fgets(s, MAX, dat))!=NULL){ /* red po red, NULL za kraj */
        c=s; zbor=vozbor=0;
        while (*c!='\n'){
            if (isalpha(*c)){
                if (!vozbor)
                    vozbor=1;
            }
            else
                if (vozbor){ /*ako do sega sme bile vo zbor-izbroj go*/
                    vozbor=0;
                    zbor++;
                }
            c++;
        }
        if (vozbor)
            zbor++; /* proveri da ne zavrsuva na bukva */
        if (zbor >= 10)
            red++;
    }

    printf("Vкупно %d redovi imaat povekje od 10 zborovi\n", red);

    fclose(dat);
    return 0;
}

```

6. Да се напише програма која за дадена текст датотека ќе ги испечати сите зборови во кои се јавуваат повеќе од две исти букви (било која буква се јавува три или повеќе пати). Да не се прави разлика помеѓу големи и мали букви. На крајот да се отпечати колку такви зборови биле пронајдени. Зборовите се составени само од букви, а меѓусебно се одделени со најмалку еден специјален знак, цифра или белина. Се смета дека во текстот нема зборови подолги од 20 знаци. Името на датотеката се задава од командна линија. (На пр. бана, текстот, Обратното, бинарната, дефинирани, Копакабана).

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define WORDLEN 21
#define LINELEN 81

int ima_poveke_od2isti(char *w){
    char *c;
    int isti;
    while(*w){

```

```

c=w+1;
isti=1;
while(*c) {
    if(tolower(*w)==tolower(*c))
        isti++;

    c++;
}
if(isti>2)
    return 1;

w++;
}
return 0;
}

int main(int argc, char *argv[]){
    char zbor[WORDLEN],c;
    FILE *f;
    int iw=0,i=0,brzb=0;

    if(argc!=2){
        printf("Upotreba: %s ime_na_datoteka\n", argv[0]);
        return -1;
    }
    if((f=fopen(argv[1], "r"))==NULL){
        printf("Ne mozam da ja najdam datotekata %s\n", argv[1]);
        return -1;
    }

    while((c=fgetc(f))!=EOF) {
        if(isalpha(c)){
            if(!iw)
                iw=1;
            zbor[i++]=c;
        }else
            if(iw){
                zbor[i]=0;
                if(ima_poveke_od2isti(zbor)){
                    puts(zbor);
                    brzb++;
                }
                zbor[i=iw=0]=0;
            }
    }

    printf("\nVкупно %d zborovi.\n",brzb);

    fclose(f);
    return 0;
}

```

7. Да се напише програма која за дадена текстуална датотека “text.txt” ќе ги изброя зборовите. Зборовите се состојат од алфаниумерички знаци и се одделени со најмалку едно празно место или знаци за интерпункција.

```

#include <stdio.h>
#include <ctype.h>

int main(){
    char c;

```

```

int brzborovi, vozbor;
FILE *dat;
brzborovi=0; vozbor=0;

if((dat=fopen("text.txt","r"))==NULL){
    fprintf(stderr,"Ne mozam da ja otvoram datotekata text.txt");
    return(-1);
}
while((c=fgetc(dat))!=EOF){ /* citaj znak po znak se do krajot */
    if (isalnum(c)){
        if (!vozbor)
            vozbor=1;
    }else
        if (vozbor) /* ako do sega sme bili vo zbor - izbroj go */
    {
        vozbor=0;
        brzborovi++;
    }
}
if (vozbor)
    brzborovi++; /* proveri da ne zavrsuva na bukva */

printf("Vкупно %d zborovi\n",brzborovi);

fclose(dat);
return 0;
}

```

8. За дадена датотека, чие име се задава од командна линија, да се испечати вкупниот број на зборови кои почнуваат и завршуваат со иста буква. Да се провери дали името на датотеката е зададено преку командна линија.

```

#include <stdio.h>
#include <ctype.h>

int main(int argc, char *argv[]){
    char c, prva, posledna;
    int brzborovi=0, vozbor=0;
    FILE *dat;

    if(argc==1){
        printf("Ne e zadadeno imeto na datotekata");
        return -1;
    }else {
        if((dat=fopen(argv[1],"r"))==NULL){
            printf("Ne mozam da ja otvoram datotekata %s",argv[1]);
            return(-1);
        }
        while((c=fgetc(dat))!=EOF){ /* citaj znak po znak se do krajot */
            if (isalpha(c)){
                if (!vozbor){
                    vozbor=1;
                    prva=c;
                }
                posledna=c; /*posledna kje bide sekoja bukva od zborot*/
            }
            else
                if (vozbor){ /* ako do sega sme bili vo zbor */
                    vozbor=0;
                    if (prva==posledna)
                        brzborovi++; /* togas proveruvame */
                }
        }
    }
}

```

```
        }
        fclose(dat);
    }

    if (vozbor && (prva==posledna))
        brzborovi++; /* za na kraj */

    printf("%d zborovi pocnuvaat i zavrsuvaat na ista bukva\n",brzborovi);
    return 0;
}
```