

PROGRAMIRANJE 1 ABC, PROGRAMIRANJE I D PRIPREMA ZA II KOLOKVIJUM (25 POENA)

POSTAVKE ZADATAKA

Izvor: Obnavljanje od prvog kolokvijuma

1. Napisati program u kojem se učitava niz a cijelih brojeva maksimalne dužine 50, a zatim se na osnovu njega formira i ispisuje niz b u kojem se svaki element duplira. Na primjer, za niz $a = [1, 2, 3]$ niz b je $[1, 1, 2, 2, 3, 3]$. Kolika je razlika između najvećeg i najmanjeg elementa niza a ("amplituda")?

2. Napisati program u kome se za učitani realan broj x i cio broj n izračunava broj s na sljedeći način: $s = \sum_{i=1}^n \frac{\sin^i x}{i!}$.

3. Razmotrimo niz Fibonaccijevih brojeva f_0, f_1, \dots ; $f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$ ($n \geq 2$). Sa tastature se učitava broj $n \geq 1$. Želimo da ispitamo istinitost relacija $\sum_{i=1}^n f_i = f_{n+2} - 1$ i $f_{n-1} f_{n+1} = f_n^2 + (-1)^n$ (za učitano n). Napisati odgovarajući program na programskom jeziku C.

4. Učitavaju se realan broj x i cio broj n . Napisati program koji izračunava sumu $s = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$.

Izvor: Zagreb, MATH,HR, Programiranje 1, 2

1. Pitagorin trougao je pravougli trougao sa stranicama čije dužine su prirodni brojevi (na primjer 3, 4, 5 ili 5, 12, 13). Napišite program koji učitava prirodan broj n i nalazi sve Pitagorine trouglove s hipotenuzom dužine n . Za svaki takav trougao treba ispisati dužine sve tri stranice u uzlaznom poretku. Ako takvih trouglova nema, treba ispisati odgovarajuću poruku.

2. Napišite program koji učitava prirodan broj n , a zatim niz od n nenegativnih cijelih brojeva. Program treba da ispiše koliko u tom nizu ima Fibonaccijevih brojeva. Za Fibonaccijeve brojeve f_i važi: $f_0 = 0, f_1 = 1$ i $f_i = f_{i-1} + f_{i-2}$ za $i \geq 2$, tako da je $f_2 = 1, f_3 = 2, f_4 = 3, f_5 = 5, f_6 = 8, f_7 = 13, f_8 = 21, \dots$. Na primjer, za $n = 6$ i učitani niz brojeva 2, 7, 13, 11, 2, 5, u tom nizu su Fibonaccijevi brojevi 2, 13, 2 i 5, pa program treba da ispiše broj 4.

3. Napišite program koji učitava prirodan broj n , pa ispisuje sumu svih prirodnih brojeva k takvih da je $n < k < 2n$ i da je $n - k$ paran broj.

4. Napišite program koji učitava prirodan broj n , pa zatim učitava n realnih brojeva i ispisuje sumu negativnih.

5. Napisati program koji učitava sa komandne linije ime datoteke i realan broj x . Program treba da ispiše sve brojeve u datoteci koji su veći od x . Možete pretpostaviti da se u datoteci nalaze podaci tipa `float`.

Napomena: Za pretvaranje stringa u broj možete koristiti funkciju `float atof(char*)`.

6. Napišite program koji učitava sa komandne linije ime datoteke i prirodan broj n . Program treba da ispiše prvih n znakova iz datoteke. Možete pretpostaviti da se u datoteci nalazi više od n znakova.

Napomena: Za pretvaranje stringa u broj možete koristiti funkciju `int atoi(char*)`.

7. Napišite funkciju s prototipom `void glas(char *s)` koja premješta slova u stringu s , tako da samoglasnici budu ispred suglasnika. Možete pretpostaviti da će se s sastojati samo od

malih slova. Na primjer, ako je funkciji poslat string "kolokvijum", onda po izlasku iz funkcije taj string treba da bude promijenjen u "ooiuklkvjum".

8. Definišite tip **kvadrat** za čuvanje podataka o jednom kvadratu u R^2 (određenom koordinatama donje lijeve tačke (x, y) i dužinom stranice a , sa stranicama koje su paralelne apscisi i ordinati), tako da bude moguća deklaracija **kvadrat k**; . Dalje, definišite funkciju koja kao argument prihvata jedan kvadrat, dok kao povratnu vrijednost vraća udaljenost središta tog kvadrata od koordinatnog početka (znači $\sqrt{(x + a/2)^2 + (y + a/2)^2}$). Zatim definišite i funkciju koja kao varijabilni argument prihvata jedan kvadrat i dva realna broja x_1 i y_1 , pa ga zamjenjuje s kvadratom jednake stranice čije se središte nalazi u tački (x_1, y_1) (znači da se (x, y) zamjenjuje sa $(x_1 - a/2, y_1 - a/2)$); funkcija ne vraća nikakvu vrijednost.

9. Data je struktura

```
struct teatar { char predstava[50], reziser[30]; int publika; };
```

Napišite program koji učitava najviše 20 takvih struktura i zatim ispisuje naziv najposjećenije predstave i ime njenog režisera.

10. Napišite funkciju koja prima pokazivač na početak niza i dužinu niza, vraća aritmetičku i geometrijsku sredinu elemenata niza. Takođe napišite i program koji kreira takav jedan niz (dužina niza je $n \leq 50$, treba je učitati sa tastature), zatim poziva gore navedenu funkciju i na kraju ispisuje tražene rezultate.

11. Napišite program koji prima dva argumenta s komandne linije – prvi argument predstavlja ime ulazne datoteke, a drugi broj znakova. Program treba da kreira datoteku pod imenom "kopija.dat" i da u nju prekopira iz datoteke navedene kao prvi argument broj znakova naveden kao drugi argument. Ukoliko je broj znakova koje treba kopirati veći od veličine ulazne datoteke koju kopiramo, tada, po završetku rada programa, ulazna datoteka i datoteka "kopija.dat" treba po svom sadržaju da budu identične.

12. Napišite funkciju s prototipom `int broj(char *s, char c)`; koja vraća broj redova u datoteci `s` koji sadrže bar jedno pojavljivanje znaka `c`. Ako datoteku nije moguće otvoriti za čitanje, funkcija treba da vrati `-1`.

13. Napisati program koji učitava prirodne brojeve $m, n \leq 20$ i realnu matricu A dimenzije $m \times n$ i nakon toga ispisuje sumu svih "ivičnih" elemenata matrice. Na primjer, u slučaju

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

treba da bude ispisano $102 (= 1 + 2 + 3 + 4 + 8 + 12 + 16 + 15 + 14 + 13 + 9 + 5)$.

14. Aritmetička progresija. Napišite program koji učitava prirodan broj $n \geq 2$ i zatim n cijelih brojeva (označimo ih sa a_1, \dots, a_n). Program treba da ispita čine li brojevi a_1, \dots, a_n aritmetičku progresiju i da odštampa odgovarajuću poruku. Ako čine, onda treba još i da odštampa sljedeći element progresije (a_{n+1}).

15. Aritmetički izraz. Napišite program koji učitava prirodan broj n , zatim niz prirodnih brojeva a_1, \dots, a_n i na kraju prirodan broj x . Program treba da ispita može li se broj x dobiti kao rezultat izraza nastalog umetanjem operacija $+$ i $-$ između brojeva a_1, \dots, a_n (u tom poretku). Na primjer, ako je $(a_1, \dots, a_4) = (6, 2, 4, 3)$ i $x = 5$ onda program treba da odštampa "x se može prikazati", jer je $5 = 6 - 2 + 4 - 3$.

Ako ne znate da riješite ovako postavljene zadatke, pretpostavite da je $n = 4$ (učitavaju se a_1, a_2, a_3, a_4, x).

Izvor: Beograd, MATF, Osnovi programiranja, Programiranje 1

1. Napisati program koji iz datoteke ulaz.txt učitava dva polinoma (stepen prvog polinoma, pa njegovi koeficijenti, počev od slobodnog člana; stepen drugog polinoma, pa njegovi koeficijenti), izračunava njihov proizvod i na standardni izlaz štampa stepen i koeficijente proizvoda.

2. Napisati funkciju void brojanje(int a[], int brojac[], int N) čiji su argumenti a i brojac celobrojni nizovi dimenzije N. Vrednosti elemenata niza a su između 0 i N - 1. Funkcija izračunava elemente niza brojac tako da je brojac[i] jednak broju pojavljivanja broja i u nizu a.

3. Napisati program pomoću kojeg se, za dati broj n, izračunava n-ti član niza $F_n = 3F_{n-1} - 2F_{n-2} + F_{n-1}F_{n-2}$, pri čemu je $F_0 = 1$ i $F_1 = 1$. U programu ne koristiti nizove.

4. Neka je dat niz X od N celih brojeva. Sastaviti funkciju koja će iz niza X izbacivati sva pojavljivanja negativnih brojeva i popunjavati ta mesta u nizu tako što će se preostali elementi niza pomerati ka početku niza. Odrediti i novu dimenziju N niza X. Npr. ulaz: N = 6, X = 0 -2 11 0 -333 0 → izlaz: N = 4, X = 0 11 0 0.

5. Napisati program koji učitava sa standardnog ulaza dve niske sa ne više od 80 karaktera u svakoj i prirodan broj k i ispisuje na standardni izlaz poruku da li se prva niska dobila cikličnim pomeranjem druge niske za k mesta. Na primer za k = 3, niska "CDEAB" se dobila cikličnim pomeranjem niske "ABCDE". Elementi niski su velika slova.

6. Sa standardnog ulaza se učitavaju četiri tačke u ravni takve da nikoje tri tačke nisu kolinearne. Tačke se zadaju parom svojih koordinata (celi brojevi). Ispitati da li te tačke određuju konveksni četvorougao i rezultat ispisati na standardni izlaz.

7. Sa standardnog ulaza se unose dve niske koje predstavljaju elemente dva skupa. Skupovi nemaju više od 20 elemenata. Napisati program koji na standardni izlaz ispisuje niske koje predstavljaju: 1. presek, 2. uniju i 3. razliku elemenata dva skupa.

8. Program sa standardnog ulaza učitava raspored 8 topova na šahovskoj tabli. Raspored se sastoji od 8 linija sa po 8 brojeva u svakoj liniji. Svaka linija odgovara jednom redu table, a svaki broj jednom polju. Broj ima vrednost 0 ako na datom polju nema topa i vrednost 1 ako na datom polju postoji top. Program treba da ispita da li je uneseni raspored validan (tj. da li je svaki učitani broj 1 ili 0 i da li ima ukupno 8 topova na tabli), kao i da odredi da li se u datom rasporedu neka dva topa tuku (topovi se tuku ukoliko se nalaze u istom redu ili istoj koloni table). Program treba da ispiše na standardnom izlazu "raspored nije validan" ukoliko ulazni podaci nisu dobri, a u suprotnom "ne tuku se" ukoliko je raspored takav da se nijedan par topova međusobno ne tuče, odn. "tuku se" ukoliko ima topova koji se tuku.

9. Neka se relacija nad nekim skupom elemenata opisuje kvadratnom matricom na sledeći način: ako je u preseku i-te vrste i j-te kolone 1, to znači da je i-ti element u relaciji sa j-tim, ako je 0 to znači da nije u relaciji. Sa standardnog ulaza zadaje se najpre dimenzija ovakve matrice, pa zatim elementi matrice, jedan za drugim, po vrstama. Dimenzija matrice ne prelazi 20. Napisati program koji, pošto proverí korektnost ulaza, za ovako zadatu relaciju ispituje njenu refleksivnost, simetričnost i tranzitivnost i odgovarajuće poruke štampa na ekran.

10. Napisati program koji za dva data pravougaonika R0 i R1 sa stranicama paralelnim koordinatnim osama izračunava i na standardni izlaz ispisuje površinu njihovog preseka ($R0 \cap R1$). Pravougaonici se učitavaju sa standardnog ulaza i zadati su koordinatama donjeg levog, odn. gornjeg desnog temena. Ove koordinate su realni brojevi. Primer: za pravougaonike zadate na sledeći način: 10 20 30 40 20 30 40 50 program treba da ispiše: Povrsina preseka iznosi 100.

11. Prvi red standardne ulazne datoteke sadrži 2 cela broja manja od 50 koji predstavljaju redom broj vrsta i broj kolona realne matrice A . Svaki sledeći red sadrži po jednu vrstu matrice. Napisati program koji nalazi i štampa sve četvorke oblika $(A(i, j), A(i + 1, j), A(i, j + 1), A(i + 1, j + 1))$ u kojima su svi elementi međusobno različiti.

12. Napisati program koji učitava sa standardnog ulaza prvo jednu liniju teksta dužine do 20, a zatim još jednu liniju do 20 sa karakterima koje treba izbaciti iz prve linije. Program treba da izbaciti specificirane karaktere iz prve linije i ispiše ono što preostane od iste. Primer: ako je unos imao sledeći oblik:

```
Hello, world!
```

```
aeiou,
```

program treba da ispiše:

```
Hll wrld!
```

13. Svaka linija datoteke čije se ime prosleđuje komandnom linijom sadrži po 6 celih brojeva: $x_1, y_1, x_2, y_2, x_3, y_3$ koji predstavljaju redom koordinate temena jednog trougla. Linija u datoteci nema više od 100. Napisati program koji će izbrojati koliko ima jednakostraničnih, odn. koliko ima pravouglih. Odgovarajuće rezultate štampati na ekran.

14. Svaki red datoteke čije se ime zadaje komandnom linijom, sadrži po 3 cela broja: A, B, C (A i B nisu istovremeno jednaki nuli), koji predstavljaju koeficijente prave u ravni $Ax + By + C = 0$. Broj redova u datoteci nije veći od 100. Napisati program koji pronalazi i na standardnom izlazu ispisuje sve parove paralelnih pravih, kao i sve trojke pravih koje se seku u jednoj tački. Način prikaza traženih podataka je proizvoljan, ali treba voditi računa o njihovoj preglednosti.

15. Sa standardnog ulaza se unosi ime datoteke čiji prvi red sadrži dimenziju celobrojne kvadratne matrice n ($n > 100$), a ostali redovi elemente matrice (vrstu po vrstu). Formirati niz b dimenzije n čiji je prvi član suma elemenata glavne dijagonale, drugi suma elemenata na prvoj donjoj dijagonalnoj paraleli (nju čine elementi odmah ispod glavne dijagonale), treći element suma druge donje dijagonalne paralele, itd. Ispisati niz na standardni izlaz.

Dolaze dva zadatka

RJEŠENJA ZADATAKA

Zadaci iz obnavljanja

① Napisati program u kojem se učitava niz a cijelih brojeva maksimalne dužine 50, a zatim se na osnovu njega formira i ispisuje niz b u kojem se svaki element duplira. Na primjer, za niz $a = [1, 2, 3]$ niz b je $[1, 1, 2, 2, 3, 3]$. Kolika je razlika između najvećeg i najmanjeg elementa niza a ("amplituda")?

Rješenje:

```
#include <stdio.h>
main(){
int a[50], b[100], n, i,
min, max, ampli;
printf("unesite n, a1, ..., an\n");
scanf("%d", &n);
for(i=0; i<=n-1; i++)
scanf("%d", &a[i]);
for(i=0; i<=n-1; i++)
{b[2*i]=a[i];
b[2*i+1]=a[i];};
printf("niz b glasi\n");
for(i=0; i<=2*n-1; i++)
printf("%d ", b[i]);
min=a[0]; max=a[0];
for(i=0; i<=n-1; i++)
{if (a[i]<min) min=a[i];
if (a[i]>max) max=a[i];};
ampli=max-min;
printf("\n amplituda = %d", ampli);}
```

Želimo da ulazni podaci glase $n = 4$, $a = [55, 66, 77, 88]$. Prilikom izvršavanja programa, imamo na ekranu:

```
unesite n, a1, ..., an
```

```
4 55 66 77 88
```

```
niz b glasi
```

```
55 55 66 66 77 77 88 88
```

```
amplituda = 33
```

Od toga, mi smo otkucali `4 55 66 77 88` (enter), a ostalo je računar saopštio.

② Napisati program u kome se za učitani realan broj x i cio broj n izračunava broj s na sljedeći način: $s = \sum_{i=1}^n \frac{\sin^i x}{i!}$.

Uputstvo: Lako se zapaža da je $s = a_1 + \dots + a_n$, gdje je $a_1 = z$, $a_i = za_{i-1}/i$ u slučaju $2 \leq i \leq n$, s tim da je $z = \sin x$.

Rješenje:

```
#include <stdio.h>
#include <math.h>
main(){ float x, z, s, sabirak;
int n, i;
printf("unesite x, n\n");
scanf("%f%d", &x, &n);
z=sin(x);
s=0;
sabirak=z;
for(i=1; i<=n; i++)
{s=s+sabirak;
sabirak=z*sabirak/(i+1);};
printf("rezultat %f", s);}
```

Prilikom izvršavanja programa, treba otkucati x i n . Nakon izvršavanja, na ekranu će stajati:

```
unesite x, n
```

```
0.52 3
```

```
rezultat 0.640771
```

To znači da je $s = 0,640771$, ako je $x = 0,52$, $n = 3$.

③ Razmotrimo niz Fibonaccijevih brojeva f_0, f_1, \dots ; $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$ ($n \geq 2$). Sa tastature se učitava broj $n \geq 1$. Želimo da ispitamo istinitost relacija $\sum_{i=1}^n f_i = f_{n+2} - 1$ i $f_{n-1}f_{n+1} = f_n^2 + (-1)^n$ (za učitano n). Napisati odgovarajući program na programskom jeziku C.

Dajemo tabelu Fibonaccijevih brojeva:

n	1	2	3	4	5	6	7	8	9	10	11	12
f_n	1	1	2	3	5	8	13	21	34	55	89	144

Rješenje:

Smisao promjenljivih $L1 = \sum_{i=1}^n f_i$, $R1 = f_{n+2} - 1$, $L2 = f_{n-1}f_{n+1}$, $R2 = f_n^2 + (-1)^n$.

```
#include <stdio.h>
main(){
int fibo[25], n, i, L1, R1, L2, R2, x;
printf("unesite n ");
scanf("%d", &n);
fibo[0]=0; fibo[1]=1;
for(i=2; i<=n+2; i++)
```

```

    fibo[i]=fibo[i-1]+fibo[i-2];
L1=0;
for(i=1; i<=n; i++)
    L1=L1+fibo[i];
printf("L1=%d ", L1);
R1=fibo[n+2]-1;
printf("R1=%d ", R1);
printf("prva rel. ");
if (L1==R1) printf("istinita\n");
    else printf("nije istinita\n");
L2=fibo[n-1]*fibo[n+1];
printf("L2=%d ", L2);
if (n%2) x=-1; else x=1;
R2=fibo[n]*fibo[n]+x;
printf("R2=%d ", R2);
printf("druga rel. ");
if (L2==R2) printf("istinita\n");
    else printf("nije istinita\n");
    Prilikom propuštanja programa, na ekranu:
unesite n 8
L1=54 R1=54 prva rel. istinita
L2=442 R2=442 druga rel. istinita
    Testiramo program sa drugim ulaznim po-
dacima:
unesite n 9
L1=88 R1=88 prva rel. istinita
L2=1155 R2=1155 druga rel. istinita

```

④ Učitavaju se realan broj x i cio broj n . Napisati program koji izračunava sumu $s = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$.

Uputstvo: Lako se zapaža da je $s = a_0 + \dots + a_n$, gdje je $a_0 = x$, $a_i = \frac{x^{2i+1}}{2i(2i+1)}$ u slučaju $1 \leq i \leq n$.

Rješenje:

```

/*probax4.c-->Digital Mars compiler*/
#include <stdio.h>
main(){ float x, s, sabirak;
int n, i;
printf("unesite x, n ");
scanf("%f%d", &x, &n);
s=x; sabirak=x;
for(i=1; i<=n; i++)
    {sabirak=sabirak*x*x/(2*i)/(2*i+1);
    s=s+sabirak;};
printf("rezultat s = %f",s);}

```

Tokom izvršavanja programa, mi ćemo otkucati x i n . Na ekranu:

```

unesite x, n 0.5 2
rezultat s = 0.521094

```

Sitni gramatički primjeri

① Rad sa stringom (rad sa niskom)

Želimo da isprobamo elementarne operacije sa niskom.

```

#include <stdio.h>
#include <string.h>
main(){
char s[10];
strcpy(s, "qwer ty");
printf("X%sY\n", s);
s[0]=97; s[2]=0;
printf("X%sY\n", s);
printf("treba string\n");
scanf("%s", s);
s[1]='\n';
/* ili svejedno s[1]=10; */
printf("X%sY\n", s);
printf("%d %d %d %d %d %d %d\n",
s[0],s[1],s[2],s[3],s[4],s[5],s[6]);}

```

Sejviramo sors kao `probas.c`. Pomoću prevodioca, kreiramo izvršni oblik programa `probas.exe`. Zatim ga pozivamo iz komandne linije tako što otkucamo

```

probas<enter>
(počinje izvršavanje). Na ekranu će pisati:
Xqwer tyY
XawY
treba string
DANAS

```

```

XD
NASY
68 10 78 65 83 0 121

```

Od toga, mi smo otkucali `DANAS<enter>`, a ostalo je računar saopštio.

Komentar. Instr. `s[2]=0`; upisuje znak za kraj niske na poziciju $i = 2$ u niski. Instrukcija `s[1]='\n'`; upisuje znak LF (line feed, new line) na poziciju $i = 1$.

q	w	e	r	␣	t	y	•	—	—
---	---	---	---	---	---	---	---	---	---

② Argumenti komandne linije

Na redu je najprostija vježba čiji je cilj samo da se savlada razmatrani pojam.

```
/* Command Line Arguments */
#include <stdio.h>
main(int argc, char *argv[])
{printf("argc = %d\n", argc);
 printf("(1) %s ", argv[1]);
 printf("(2) %s ", argv[2]);
 printf("(3) %s ", argv[3]);}
```

Sejviramo kao probaa.c. Prevodenjem dobijamo probaa.exe. Pozivamo sa probaa 1234 qwer as x x<enter>

Na ekranu:

```
argc = 6
(1) 1234 (2) qwer (3) as
```

Dopuna o atoi i atof.

Često je potrebno da se preko komandne linije prenese brojna vrijednost. Za te svrhe služe funkcije atoi(s) čija je vrijednost tipa int i atof(s) čija je vrijednost float, pod uslovom da je s string. Evo primjera:

```
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[])
{int m, n; float p;
 printf("argc = %d\n", argc);
 m = atoi(argv[1]);
 n = atoi(argv[2]);
 p = atof(argv[3]);
 printf("%d %d %d %f %f",
 m, n, m+n, p, p*p);}
```

Neka se zove probaax.c. Kreiramo izvršni fajl probaax.exe. Puštamo da radi sa probaax 21 -2 7.4<enter>

Time dobijamo na ekranu:

```
argc = 4
21 -2 19 7.400000 54.760001
```

Primjeri za command prompt:

```
C:\windows>
C:\clangua\bin>
```

Primjeri komandne linije:

```
C:\clangua\bin>dir
C:\clangua\bin>probas
```

Sa argumentima:

```
C:\clangua\bin>dir *.exe
C:\clangua\bin>probaax 21 -2 7.4
```

③ Rad sa fajlom (rad sa datotekom, čitanje)

```
/* probad.c */
#include <stdio.h>
main()
{FILE* fp; int x, i, a; char c;
 fp = fopen("probaxxx.txt", "r");
 x = fp!=NULL;
 printf("open x=%d\n", x);
 for(i=1; i<=3; i++)
 {c = fgetc(fp);
 a = feof(fp)!=0;
 printf("znak %c(%d) kraj %d\n",
 c, c, a);};
 fclose(fp);}
```

Uzmimo da (u trenutku pozivanja programa) fajl probaxxx.txt zuzima 0bytes. Tada ćemo na ekranu imati (kao rezultat rada)

```
open x=1
znak (-1) kraj 1
znak (-1) kraj 1
znak (-1) kraj 1
```

Ako glasi p i (znači) zauzima 1byte onda ćemo na ekranu dobiti

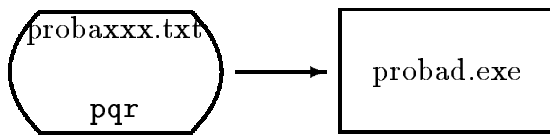
```
open x=1
znak p(112) kraj 0
znak (-1) kraj 1
znak (-1) kraj 1
```

Ako glasi pqr (3bytes) onda

```
open x=1
znak p(112) kraj 0
znak q(113) kraj 0
znak r(114) kraj 0
```

U slučaju da (u trenutku pozivanja) u tekućem folderu nema fajla sa takvim nazivom, imaćemo na ekranu

```
open x=0
i program "iskače", operativni sistem šalje poruku "illegal operation".
```



④ Rad sa fajlom (upisivanje)

```

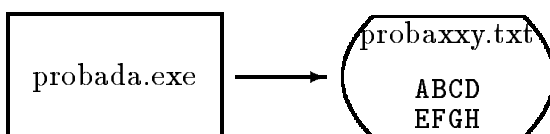
/* probada.c */
#include <stdio.h>
main(){ FILE* fp;
int i, j; char c;
printf("Otvaranje za write\n");
fp = fopen("probaxxy.txt", "w");
/* fp - file pointer */
i = 65; /* char. A */
while (i<69)
    {fputc(i, fp); i++;}
j = '\n'; /* new line */
/* ili svejedno j = 10; */
fputc(j, fp);
while (i<73)
    {fputc(i, fp); i++;}
fputc(j, fp);
fclose(fp);
printf("Fajl je zatvoren\n");
printf("Sad read slovo po slovo\n");
fp = fopen("probaxxy.txt", "r");
for(i=1; i<=14; i++)
    {c = fgetc(fp);
    printf("(%d)%c%d ", i, c, c);}}
  
```

Prilikom izvršavanja, na ekranu se pojavljuje:

```

Otvaranje za write
Fajl je zatvoren
Sad read slovo po slovo
(1)A65 (2)B66 (3)C67 (4)D68 (5)
10 (6)E69 (7)F70 (8)G71 (9)H72 (10)
10 (11) -1 (12) -1 (13) -1 (14) -1
  
```

Pored toga, na disku će biti prisutan novi fajl probaxxy.txt (12bytes), čiji sadržaj glasi:
 ABCD
 EFGH



Napomena o softveru za programski jezik C

Da bismo propuštali programe (da bismo vježbali), potrebno je da nabavimo i instaliramo prevodilac za C. Preporučuje se softverski paket Code::Blocks. Takođe se preporučuje i Digital Mars, radi u command prompt-u.

O upotrebi, u slučaju command prompt-a.

Ako se pozove command prompt onda se ekran nalazi u tekstualnom režimu i korisnik upućuje jednu po jednu komandu operativnom sistemu, s tim da pojedina komanda zauzima jedan red na ekranu.

Pretpostavimo da smo instalirali softver u folder c:\clangua. Tada je c:\clangua\bin radni folder, radni direktorijum.

Uzmimo kao primjer program "probas". Mi znamo da ukupno postoje tri etape rada.

(1) Editovanje programa, viši se pomoću nekog tekstualnog editora, ascii editora. Može da posluži Crimson Editor (iz posebnog prozora). Zapisati sors pod nazivom probas.c u radni folder. Osvjedočiti se da je fajl pod tim nazivom prisutan u tom folderu i da zauzima (recimo) 282 bytes.

(2) Prevođenje programa. Treba se pozicionirati u već spomenuti radni folder. Iz komandne linije kažemo računaru (otkucamo)

```
sc probas<enter>
```

Imaćemo fajl probas.exe, to je tzv. izvršni oblik programa. Osvjedočiti se da je taj fajl prisutan i da zauzima (recimo) 45084 bytes.

(3) Izvršavanje. I dalje se nalazimo u folderu c:\clangua\bin. Obično se kaže "tekući folder" za onaj u kome se trenutno nalazimo. Treba otkucati

```
probas<enter>
```

(program se poziva po svom imenu); što očito predstavlja primjer komandne linije. Kaže se da se program izvršava iz operativnog sistema (OS-a). Srazmjerno potrebi, unosimo ulazne podatke, a računar nam saopštava rezultate.

