1. Što će se ispisati izvršavanjem sljedećih naredbi? Zadatak riješiti "na papiru", a zatim provjeriti rješenje izvršavanjem odsječka programa na vlastitom računalu. Napomena: u svim ovdje prikazanim naredbama `0` predstavlja znamenku nula, a ne slovo O.

```
char c;
c = 'A' + '0';
printf("%d\n", c);
printf("%c\n", c);
printf("%c\n", 'D' - 'A' + '0');
printf("%d\n", 'D' - 'A' + '0');
printf("%d\n", '7' - '5');
printf("%d\n", '7' - 5);
printf("%c\n", '7' - 5);
printf("%d\n", '0' % 10);
```

2. S tipkovnice učitati niz znakova (string) koji zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 20 znakova. Ako je u niz učitana i oznaka novog retka, izbaciti je iz učitanog niza. Ispisati niz znakova. Pronaći i ispisati najveću ASCII vrijednost znaka u nizu.

Primjeri izvršavanja programa

```
Upisite niz > Ispisati niz znakova↵
Niz: Ispisati niz znakova
Najveca ASCII vrijednost: 122
```

```
Upisite niz > Zar doista ponovo ispisati niz znakova↵
Niz: Zar doista ponovo is
Najveca ASCII vrijednost: 118
```

3. S tipkovnice učitati niz znakova (string) koji zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 8 znakova. Ako je u niz učitana i oznaka novog retka, izbaciti je iz učitanog niza. Ispisati niz znakova. Ako se u nizu nalazi ispravan heksadekadski broj (znamenke tog broja mogu biti samo znamenkama 0-9 ili velikim slovima A-F) izračunati i ispisati ekvivalentan dekadski broj (po tipu unsigned int). Ako učitani niz ne predstavlja ispravan heksadekadski broj, ispisati prvi znak i poziciju znaka zbog kojeg niz ne predstavlja ispravan heksadekadski broj.

Primjeri izvršavanja programa

```
Upisite niz > AF12aDD↵
Niz: AF12aDD↵
Neispravan znak a na poziciji 5.
```

```
Upisite niz > 7FFFFFFFxxxx↵
Niz: 7FFFFFFF
Dekadski: 2147483647
```

```
Upisite niz > FFFFFFFF↵
Niz: FFFFFFFF
Dekadski: 4294967295
```

4. S tipkovnice učitati nenegativni cijeli broj (unsigned int). Načiniti niz znakova (string) koji sadrži heksadekadske znamenke ekvivalentnog broja. Za znamenke koristiti znamenke 0-9 i velika slova A-F. Ispisati niz.

Primjeri izvršavanja programa

```
Upisite dekadski broj > 0↵
Heksadekadski: 0
```

```
Upisite dekadski broj > 26↵
Heksadekadski: 1A↵
```

```
Upisite dekadski broj > 4294967295↵
Heksadekadski: FFFFFFFF
```

5. S tipkovnice učitati niz znakova (string) koji zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 20 znakova. Ako je u niz učitana i oznaka novog retka, izbaciti je iz učitanog niza. Ispisati niz znakova. Izračunati i ispisati koliko u učitanom nizu ima samoglasnika, koliko suglasnika, a koliko ostalih znakova.

Primjer izvršavanja programa

```
Upisite niz > printf("%s", "Ana");↵
Niz: printf("%s", "Ana");↵
Samoglasnika: 3↵
Suglasnika: 7↵
Ostalih: 10
```

6. S tipkovnice, u obliku cijelih dekadskih brojeva, učitati najviše 8 ASCII vrijednosti znakova. Učitavanje prekinuti i onda kada korisnik unese ASCII vrijednost koja ne predstavlja veliko slovo abecede. Učitane vrijednosti pohraniti u polje znakova koje se koristi kao niz znakova. Na kraju, ispisati dobiveni niz znakova pomoću funkcije printf i formata %s

Primjeri izvršavanja programa

```
Upisite brojeve > 80 69 82 85 64 80 69↵
PERU↵
```

```
Upisite brojeve > 65 82 71 69 78 84 73 78 65 66↵
ARGENTIN↵
```

7. Koje vrijednosti, izraženo dekadski, će sadržavati varijable a, b nakon izvršavanja sljedećeg programskog odsječka:

```
char a;
short int b;
a = 120;
b = 32000;
a = a + 10;
b = b + 1000;
```

Svoje rješenje provjerite tako da programski odsječak, dopunjen naredbom za ispis vrijednosti varijabli na zaslon i ostalim potrebnim naredbama, izvršite na svom računalu.

8. S tipkovnice učitati niz znakova (string) koji zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 100 znakova. Ako je u niz učitana i oznaka novog retka, izbaciti je iz učitanog niza. Učitati cijeli broj te ako broj predstavlja ispravnu poziciju znaka u nizu, iz niza izbaciti znak na zadanoj poziciji te promijenjeni niz ispisati na zaslon. Inače, ispisati poruku "Neispravna pozicija". Smatra se da pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

   Primjeri izvršavanja programa

```
Upisite niz > Ovo je niz↵
Upisite poziciju > 0↵
Neispravna pozicija
```

```
Upisite niz > Ovo je niz↵
Upisite poziciju > 3↵
Ov je niz
```

9. S tipkovnice učitati niz znakova (string) koji zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 100 znakova. Ako je u niz učitana i oznaka novog retka, izbaciti je iz učitanog niza. Učitati cijeli broj te ako broj predstavlja ispravnu poziciju znaka u nizu, u niz na zadanu poziciju ubaciti znak X te promijenjeni niz ispisati na zaslon. Inače, ispisati poruku "Neispravna pozicija". Smatra se da pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

   Primjer izvršavanja programa

```
Upisite niz > Ovo je niz↵
Upisite poziciju > 3↵
OvXo je niz
```

10. S tipkovnice učitati dva niza znakova (string) od kojih svaki zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 20 znakova. Ako je u neki od nizova učitana i oznaka novog retka, izbaciti je iz tog niza. Učitati cijeli broj te ako broj predstavlja ispravnu poziciju znaka u prvom nizu, u prvi niz na zadanu poziciju umetnuti drugi niz te ispisati (sada promijenjeni) prvi niz na zaslon. Inače, ispisati poruku "Neispravna pozicija". Smatra se da pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

    Primjeri izvršavanja programa

```
Upisite 1. niz > Ovo je niz↵
Upisite 2. niz > Novi niz↵
Upisite poziciju > 12↵
Neispravna pozicija
```

```
Upisite 1. niz > Ovo je niz↵
Upisite 2. niz > Novi niz↵
Upisite poziciju > 3↵
OvNovi nizo je niz
```

11. S tipkovnice učitati dva niza znakova (string) od kojih svaki zajedno s eventualno učitanom oznakom novog reda sigurno neće biti dulji od 20 znakova. Ako je u neki od nizova učitana i oznaka novog retka, izbaciti je iz tog niza. U treći niz znakova (string) upisati rezultat operacije nadovezivanja učitanih nizova (konkatenacije nizova) te ispisati treći niz.

    Primjer izvršavanja programa

```
Upisite 1. niz > Ovo je niz↵
Upisite 2. niz > Novi niz↵
Ovo je nizNovi niz
```

12. S tipkovnice učitati niz znakova (string) koji zajedno s eventualnom učitanom oznakom novog reda sigurno neće biti dulji od 80 znakova. Za svako malo slovo engleske abecede (znakovi a-z) koje se barem jednom pojavilo u učitanom nizu, ispisati koliko se puta pojavilo u učitanom nizu.

    Primjer izvršavanja programa

```
Upisite niz > Da, nenadoknadivo↵
slovo 'a' pojavilo se  3 puta
slovo 'd' pojavilo se  2 puta
slovo 'e' pojavilo se  1 puta
slovo 'i' pojavilo se  1 puta
slovo 'k' pojavilo se  1 puta
slovo 'n' pojavilo se  3 puta
slovo 'o' pojavilo se  2 puta
slovo 'v' pojavilo se  1 puta
```

13. Napisati sadržaj registra u kojem je, prema IEEE 754 standardu za prikaz brojeva u standardnoj (*single*) preciznosti, pohranjen broj -17.78125_{10}. Sadržaj registra napisati u oktalnom i heksadekadskom obliku.

14. U registru od 32 bita upisan je broj $C2\ B0\ 00\ 00_{16}$. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

15. U registru od 32 bita upisan je broj $43\ 00\ 20\ 00_{16}$. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

16. U registru od 32 bita upisan je broj $3\ 01\ 22\ 40\ 00\ 00_{8}$. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

17. U registru od 32 bita upisan je broj $3\ 77\ 40\ 00\ 00\ 00_{8}$. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

18. U registru od 32 bita upisan je broj $7F\ C0\ 00\ 00_{16}$. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

19. U registru od 32 bita upisan je broj 80 00 00 00_{16}. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

20. U registru od 32 bita upisan je broj 00 68 00 00_{16}. Napisati koji je broj predstavljen u tom registru, ako registar služi za pohranu varijable tipa *float*. Rezultat napisati u dekadskom brojevnom sustavu.

21. S tipkovnice učitati sljedeće podatke za 10 osoba: ime (niz znakova koji nije dulji od 30 znakova), prezime (niz znakova koji nije dulji od 30 znakova). Podaci se upisuju na način prikazan u primjeru. Nakon učitavanja podataka, podatke o osobama ispisati u obrnutom redoslijedu u odnosu na redoslijed učitavanja.

Primjer izvršavanja programa

```
 1. Ime      > Ana Marija↵
 1. Prezime  > Horvat↵
 2. Ime      > Petar↵
 2. Prezime  > Novak↵
 ... itd.
10. Ime      > Ivana↵
10. Prezime  > Kolar Ban↵
↵
Kolar Ban, Ivana↵
 ... itd.
Novak, Petar↵
Horvat, Ana Marija↵
```

22. Učitati cijeli broj *n*. Nakon toga za *n* osoba učitati matični broj (cijeli broj) i godinu rođenja osobe (cijeli broj). Nakon što su podaci za sve osobe učitani, ispisati ih poredane prema godini rođenja (od manjih prema većim), a unutar godine rođenja poredano prema matičnom broju (od manjih prema većim).

Primjer izvršavanja programa

```
Upisite broj osoba > 11↵
  1. osoba > 200100 1989↵
  2. osoba > 100107 1985↵
  3. osoba > 200102 1989↵
  4. osoba > 200105 1989↵
  5. osoba > 300300 1992↵
  6. osoba > 100305 1992↵
  7. osoba > 300105 1989↵
  8. osoba > 100106 1991↵
  9. osoba > 100150 1992↵
 10. osoba > 200101 1992↵
 11. osoba > 100202 1985↵
↵
1985. godina↵
   100107↵
   100202↵
1989. godina↵
   200100↵
   200102↵
   200105↵
   300105↵
1991. godina↵
   100106↵
1992. godina↵
   100150↵
   100305↵
   200101↵
   300300↵
```

23. Učitati cijeli broj *n*. Nakon toga za *n* oglasa za prodaju nekretnina učitati šifru oglasa (cijeli broj), cijenu nekretnine (cijeli broj) i površinu nekretnine (cijeli broj). Ispisati oglase poredane po cijeni (od većih prema manjim), unutar toga poredano po površini (od većih prema manjim), a unutar toga poredano prema šifri oglasa (od manjih prema većim).

Primjer izvršavanja programa

```
Upisite broj oglasa > 12↵
  1. oglas > 43000  500000  600↵
  2. oglas > 512    500000  600↵
  3. oglas > 984    100000  200↵
  4. oglas > 52415 1100000 2000↵
  5. oglas > 10001  500000  450↵
  6. oglas > 10005  100000  175↵
  7. oglas > 30002  500000  600↵
  8. oglas > 20007 1100000 1800↵
  9. oglas > 30001  500000  350↵
 10. oglas > 30350  100000  185↵
 11. oglas > 1002  1100000 2000↵
 12. oglas > 60000  750000  950↵
↵
 1100000 EUR     2000 m2     1002↵
 1100000 EUR     2000 m2    52415↵
 1100000 EUR     1800 m2    20007↵
  750000 EUR      950 m2    60000↵
  500000 EUR      600 m2      512↵
  500000 EUR      600 m2    30002↵
  500000 EUR      600 m2    43000↵
  500000 EUR      450 m2    10001↵
  500000 EUR      350 m2    30001↵
  100000 EUR      200 m2      984↵
  100000 EUR      185 m2    30350↵
  100000 EUR      175 m2    10005↵
```

24. **Rješenja:**

1. -

2.
```
#include <stdio.h>
#define MAX_NIZ 20

int main(void) {
   char niz[MAX_NIZ + 1];

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   // izbaci znak novog retka
   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n')
         niz[i] = '\0';
      i = i + 1;
   }
   printf("Niz: %s\n", niz);

   int najveci = niz[0];
   i = 0; // krece od nule jer bi niz mogao biti prazan
   while (niz[i] != '\0') {
      if (niz[i] > najveci) {
         najveci = niz[i];
      }
      i = i + 1;
   }
   printf("Najveca ASCII vrijednost: %d", najveci);

   return 0;
}
```

3.
```
#include <stdio.h>
#define MAX_NIZ 8

int main(void) {
   char niz[MAX_NIZ + 1];

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   // izbaci znak novog retka
   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n')
         niz[i] = '\0';
      i = i + 1;
   }
   printf("Niz: %s\n", niz);

   unsigned int broj = 0U;

   i = 0;
   while (niz[i] != '\0' && ((niz[i] >= '0' && niz[i] <= '9') ||
                             (niz[i] >= 'A' && niz[i] <= 'F'))) {
      if (niz[i] >= '0' && niz[i] <= '9') {
         broj = broj * 16 + niz[i] - '0';
      } else {
         broj = broj * 16 + niz[i] - 'A' + 10;
      }
      i = i + 1;
   }
   if (niz[i] == '\0') {
      // doslo se do kraja, dakle sve znamenke su bile ispravne
      printf("Dekadski: %u", broj);
   } else {
      printf("Neispravan znak %c na poziciji %d.", niz[i], i + 1);
   }

   return 0;
}
```

4.
```
#include <stdio.h>
#define MAX_NIZ 8

int main(void) {
   char znamenke[MAX_NIZ]; // 1-d polje obrnuto poredanih znamenki
   char niz[MAX_NIZ + 1];  // niz u koji ce se prepisati znamenke iz 1-d polja
   unsigned int broj;
   int broj_znamenki = 0;

   printf("Upisite dekadski broj > ");
   scanf("%u", &broj);

   if (broj == 0) {
      znamenke[0] = '0';
      broj_znamenki = broj_znamenki + 1;
   } else {
      int ostatak;
      while (broj != 0) {
         ostatak = broj % 16;
         broj = broj / 16;
         if (ostatak >= 10) {
            znamenke[broj_znamenki] = ostatak - 10 + 'A';
         } else {
            znamenke[broj_znamenki] = ostatak + '0';
         }
         broj_znamenki = broj_znamenki + 1;
      }
   }

   // prepisi polje znamenke u niz, obrnutim redom
   int i;
   for (i = 0; i < broj_znamenki; i = i + 1) {
      niz[i] = znamenke[broj_znamenki - i - 1];
   }
   // obavezno terminirati niz
   niz[broj_znamenki] = '\0';

   printf("Heksadekadski: %s", niz);

   return 0;
}
```

5.
```c
#include <stdio.h>
#define MAX_NIZ 20

int main(void) {
   char niz[MAX_NIZ + 1];

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   // izbaci znak novog retka
   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n')
         niz[i] = '\0';
      i = i + 1;
   }
   printf("Niz: %s\n", niz);
   int samog = 0, sug = 0, ostali = 0;
   i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == 'A' || niz[i] == 'E' || niz[i] == 'I' || niz[i] == 'O' ||
          niz[i] == 'U' || niz[i] == 'a' || niz[i] == 'e' || niz[i] == 'i' ||
          niz[i] == 'o' || niz[i] == 'u') {
         samog = samog + 1;
      } else if ((niz[i] >= 'A' && niz[i] <= 'Z') ||
                 (niz[i] >= 'a' && niz[i] <= 'z')) {
         sug = sug + 1;
      } else {
         ostali = ostali + 1;
      }
      i = i + 1;
   }

   printf("Samoglasnika: %d\n", samog);
   printf("Suglasnika: %d\n", sug);
   printf("Ostalih: %d\n", ostali);

   return 0;
}
```

6.
```c
#include <stdio.h>
#define MAX_NIZ 8

int main(void) {
   char niz[MAX_NIZ + 1];
   int ascii, i = 0;
   printf("Upisite brojeve > ");

   do {
      scanf("%d", &ascii);
      if (ascii >= 'A' && ascii <= 'Z') {
         niz[i] = ascii;
         i = i + 1;
      }
   } while (i < MAX_NIZ && ascii >= 'A' && ascii <= 'Z');

   // zasto je sljedeca naredba vazna?
   niz[i] = '\0';

   printf("%s\n", niz);

   return 0;
}
```

7.
```c
#include <stdio.h>

int main(void) {
   char a;
   short int b;
   a = 120;
   b = 32000;
   a = a + 10;
   b = b + 1000;

   printf("%hhd %hd", a, b);

   return 0;
}
```

8.
```
#include <stdio.h>
#define MAX_NIZ 100

int main(void) {
   char niz[MAX_NIZ + 1];

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   // izbaci znak novog retka
   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n')
         niz[i] = '\0';
      i = i + 1;
   }
   int duljina = 0;
   while (niz[duljina] != '\0') {
      duljina = duljina + 1;
   }

   int poz;
   printf("Upisite poziciju > ", &poz);
   scanf("%d", &poz);

   if (poz >= 1 && poz <= duljina) {
      /* Sve znakove od indeksa poz-1 do indeksa dulj
         posmaknuti za jedan u lijevo. Uocite, posmaknut ce se
         i \0, tako da ce niz na kraju biti kraci za jedan znak */
      for (i = poz - 1; i < duljina; i = i + 1) {
         niz[i] = niz[i + 1];
      }
      printf("%s", niz);
   } else {
      printf("Neispravna pozicija");
   }
   return 0;
}
```

alternativno, skraćeno, jednako ispravno
(slično skraćenje se može, a ne mora, primijeniti i na drugim zadacima)

```
   // izbaci znak novog retka i izracunaj duljinu niza bez \n
   int duljina = 0, i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n') {
         niz[i] = '\0';
      } else {
         duljina = duljina + 1;
      }
      i = i + 1;
   }
```

9.
```c
#include <stdio.h>
#define MAX_NIZ 100

int main(void) {
   char niz[MAX_NIZ + 1];

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   // izbaci znak novog retka
   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] == '\n')
         niz[i] = '\0';
      i = i + 1;
   }
   int duljina = 0;
   while (niz[duljina] != '\0') {
      duljina = duljina + 1;
   }

   int poz;
   printf("Upisite poziciju > ", &poz);
   scanf("%d", &poz);

   if (poz >= 1 && poz <= duljina) {
      /* Sve znakove od indeksa poz-1 do indeksa duljina
         posmaknuti za jedan u desno, ali mora se zapoceti
         od zadnjeg znaka. Uocite, posmaknut ce se
         i \0, tako da ce niz na kraju biti dulji za jedan znak */
      for (i = duljina; i >= poz - 1; i = i - 1)
         niz[i + 1] = niz[i];

      /* na poziciju poz, odnosno indeks poz-1 upisi slovo X */
      niz[poz - 1] = 'X';

      printf("%s", niz);
   } else {
      printf("Neispravna pozicija");
   }

   return 0;
}
```

10.
```c
#include <stdio.h>
#define MAX_NIZ 20

int main(void) {
   char niz1[MAX_NIZ + 1 + MAX_NIZ]; // uociti + MAX_NIZ
   char niz2[MAX_NIZ + 1];

   printf("Upisite 1. niz > ");
   fgets(niz1, MAX_NIZ + 1, stdin);

   int i = 0;
   while (niz1[i] != '\0') {
      if (niz1[i] == '\n')
         niz1[i] = '\0';
      i = i + 1;
   }
   int duljina1 = 0;
   while (niz1[duljina1] != '\0') {
      duljina1 = duljina1 + 1;
   }

   printf("Upisite 2. niz > ");
   fgets(niz2, MAX_NIZ + 1, stdin);

   i = 0;
   while (niz2[i] != '\0') {
      if (niz2[i] == '\n')
         niz2[i] = '\0';
      i = i + 1;
   }
   int duljina2 = 0;
   while (niz2[duljina2] != '\0') {
      duljina2 = duljina2 + 1;
   }

   int poz;
   printf("Upisite poziciju > ", &poz);
   scanf("%d", &poz);

   if (poz >= 1 && poz <= duljina1) {
      /* Sve znakove u niz1 od indeksa poz-1 do indeksa duljina1
         posmaknuti za duljina2 u desno, ali mora se zapoceti
         od zadnjeg znaka. Uociti, posmaknut ce se
         i \0, tako da ce niz1 na kraju biti dulji za duljina2 znakova */
      for (i = duljina1; i >= poz - 1; i = i - 1)
         niz1[i + duljina2] = niz1[i];
      /* od pozicije poz, odnosno indeksa poz-1 u niz1
         upisati sadrzaj niza niz2  */
      for (i = 0; i < duljina2; i = i + 1) {
         niz1[poz - 1 + i] = niz2[i];
      }

      printf("%s", niz1);
   } else {
      printf("Neispravna pozicija");
   }

   return 0;
}
```

11.
```c
#include <stdio.h>
#define MAX_NIZ 20

int main(void) {
   char niz1[MAX_NIZ + 1];
   char niz2[MAX_NIZ + 1];
   char niz3[MAX_NIZ + 1 + MAX_NIZ]; // uociti + MAX_NIZ

   printf("Upisite 1. niz > ");
   fgets(niz1, MAX_NIZ + 1, stdin);

   int i = 0;
   while (niz1[i] != '\0') {
      if (niz1[i] == '\n')
         niz1[i] = '\0';
      i = i + 1;
   }

   printf("Upisite 2. niz > ");
   fgets(niz2, MAX_NIZ + 1, stdin);

   i = 0;
   while (niz2[i] != '\0') {
      if (niz2[i] == '\n')
         niz2[i] = '\0';
      i = i + 1;
   }

   /* prepisati niz1 u niz3 */
   int ind3;
   i = ind3 = 0;
   while (niz1[i] != '\0') {
      niz3[ind3] = niz1[i];
      ind3 = ind3 + 1;
      i = i + 1;
   }

   /* dopisati niz2 na kraj niza niz3 */
   i = 0;
   while (niz2[i] != '\0') {
      niz3[ind3] = niz2[i];
      ind3 = ind3 + 1;
      i = i + 1;
   }

   /* na kraj niza niz3 upisati \0 */
   niz3[ind3] = '\0';

   printf("%s", niz3);

   return 0;
}
```

12.
```
#include <stdio.h>
#define MAX_NIZ 20
#define DG 'a'
#define GG 'z'

int main(void) {
   char niz[MAX_NIZ + 1];
   int brojac[GG - DG + 1] = {0};

   printf("Upisite niz > ");
   fgets(niz, MAX_NIZ + 1, stdin);

   int i = 0;
   while (niz[i] != '\0') {
      if (niz[i] >= DG && niz[i] <= GG) {
         brojac[niz[i] - DG] = brojac[niz[i] - DG] + 1;
      }
      i = i + 1;
   }
   for (i = 0; i < GG - DG + 1; i = i + 1) {
      if (brojac[i] > 0) {
         printf("slovo '%c' pojavilo se %2d puta\n", i + DG, brojac[i]);
      }
   }

   return 0;
}
```

13. Prvi bit za predznak se postavlja na P=1. Time je pitanje predznaka riješeno (upamtiti: u IEEE 754 formatu se ne koristi ništa što podsjeća na tehniku dvojnog komplementa!)

Sada treba odrediti karakteristiku i mantisu. Prvo pretvoriti broj u binarni oblik:

$17.78125_{10} = 10001.11001_2$

Normalizirati:

$10001.11001_2 = 1.000111001 \cdot 2^4$

$BE = 4_{10} \Rightarrow K = 4 + 127 = 131_{10} = 10000011_2$

$M = 1.000111001$

U 32-bitni registar prepisati P, K i M (ali BEZ tzv. SKRIVENOG BITA!):

$1\ 10000011\ 00011100100000000000000$

Grupirati po tri znamenke **s desna na lijevo**:

$11\ 000\ 001\ 100\ 011\ 100\ 100\ 000\ 000\ 000\ 000_2 = 30143440000_8$

Grupirati po četiri znamenke **s desna na lijevo**:

$1100\ 0001\ 1000\ 1110\ 0100\ 0000\ 0000\ 0000_2 = C18E4000_{16}$

14. Varijable tipa float pohranjuju se prema IEEE 754 formatu standardne preciznosti

    $C2\ B0\ 00\ 00_{16} = 1100\ 0010\ 1011\ 0000\ 0000\ 0000\ 0000\ 0000_2$

    Odrediti predznak: P = 1, stoga je broj negativan.
    Odrediti binarni eksponent:

    $K = 10000101_2 = 133_{10} \Rightarrow BE = 133 - 127 = 6$

    Odrediti mantisu (vratiti joj skriveni bit):

    Mantisa bez skrivenog bita: $.01100000000000000000000_2$

    $M = 1.01100000000000000000000_2$

    Rezultat se dobije množenjem mantise s 2^{BE}:

    $1.01100000000000000000000_2 \cdot 2^6 = 1011000.0_2 = 88.0_{10}$

    Ne zaboraviti negativni predznak (jer P=1)
    Konačni rezultat: -88.0

15. 128.125_{10}

16. -12.625_{10}

17. K = 255, u mantisi su svi bitovi postavljeni na nulu. Radi se o prikazu beskonačnosti. Budući da je predznak P=1, konačno rješenje jest: $-\infty$.

18. K = 255, a u mantisi postoji jedan ili više bitova koji su postavljeni na jedinicu. U registru je prikazana vrijednost NaN (Not a Number).

19. K = 0, a u mantisi su svi bitovi postavljeni na 0. Radi se o prikazu broja 0. Budući da je predznak P=1, konačno rješenje jest: -0.0.

20. K = 0, a u mantisi postoje bitovi koji su postavljeni na jedan. Radi se o prikazu denormaliziranog broja.

    $00\ 68\ 00\ 00_{16} = 0000\ 0000\ 0110\ 1000\ 0000\ 0000\ 0000\ 0000_2$

    $BE = -126_{10}$ (kod denormaliziranog broja ne koristi se formula BE = K - 127)

    $M = 0.1101_2$ (kod denormaliziranog broja skriveni bit nije 1, nego 0)

    Rezultat se dobije množenjem mantise s 2^{BE}:

    $0.1101_2 \cdot 2^{-126} = 0.8125_{10} \cdot 2^{-126} \approx 9.55 \cdot 10^{-39}$

21.
```c
#include <stdio.h>

#define MAXNIZ 30
#define MAXOSOBA 10

int main(void) {
   struct osoba_s {
      char ime[MAXNIZ + 1];
      char prezime[MAXNIZ + 1];
   };

   struct osoba_s osoba[MAXOSOBA];

   // ucitavanje podataka
   int i;
   for (i = 0; i < MAXOSOBA; i = i + 1) {
      printf("%2d. Ime      > ", i + 1);
      fgets(osoba[i].ime, MAXNIZ + 1, stdin);
      int k = 0;
      while (osoba[i].ime[k] != '\0') {
         if (osoba[i].ime[k] == '\n') {
            osoba[i].ime[k] = '\0';
         }
         k = k + 1;
      }

      printf("%2d. Prezime  > ", i + 1);
      fgets(osoba[i].prezime, MAXNIZ + 1, stdin);
      k = 0;
      while (osoba[i].prezime[k] != '\0') {
         if (osoba[i].prezime[k] == '\n') {
            osoba[i].prezime[k] = '\0';
         }
         k = k + 1;
      }
   }

   // ispis
   printf("\n");
   for (i = MAXOSOBA  - 1; i >= 0; i = i - 1) {
      printf("%s, %s\n", osoba[i].prezime, osoba[i].ime);
   }

   return 0;
}
```

22.
```c
#include <stdio.h>

int main(void) {
   struct osoba_s {
      int mbr;
      int godRod;
   };

   int n;
   printf("Upisite broj osoba > ");
   scanf("%d", &n);

   struct osoba_s osoba[n];

   // ucitavanje podataka o osobama
   int i;
   for (i = 0; i < n; i = i + 1) {
      printf("%2d. osoba > ", i + 1);
      scanf("%d %d", &osoba[i].mbr, &osoba[i].godRod);
   }

   // sortiranje
   int ind_min, j;
   struct osoba_s pomocni;
   for (i = 0; i < n - 1; i = i + 1) {
      ind_min = i + 1;
      for (j = i + 2; j < n; j = j + 1) {
         if (osoba[j].godRod < osoba[ind_min].godRod ||
             (osoba[j].godRod == osoba[ind_min].godRod &&
              osoba[j].mbr < osoba[ind_min].mbr)
            ) {
            ind_min = j;
         }
      }
      if (osoba[ind_min].godRod < osoba[i].godRod ||
          (osoba[ind_min].godRod == osoba[i].godRod &&
           osoba[ind_min].mbr < osoba[i].mbr)) {
         pomocni = osoba[i];
         osoba[i] = osoba[ind_min];
         osoba[ind_min] = pomocni;
      }
   }

   // ispis
   printf("\n");
   int ispisujeSeGodina = -1;
   for (i = 0; i < n; i = i + 1) {
      if (osoba[i].godRod != ispisujeSeGodina) {
         printf("%4d. godina\n", osoba[i].godRod);
         ispisujeSeGodina = osoba[i].godRod;
      }
      printf("%9d\n", osoba[i].mbr);
   }

   return 0;
}
```

23.
```c
#include <stdio.h>

int main(void) {
   struct oglas_s {
      int sifOglas;
      int cijena;
      int povrsina;
   };

   int n;
   printf("Upisite broj oglasa > ");
   scanf("%d", &n);

   struct oglas_s oglas[n];

   // ucitavanje podataka o oglasima
   int i;
   for (i = 0; i < n; i = i + 1) {
      printf("%2d. oglas > ", i + 1);
      scanf("%d %d %d", &oglas[i].sifOglas, &oglas[i].cijena, &oglas[i].povrsina);
   }

   // sortiranje
   int ind_raniji;  // element (iz ostatka polja) koji treba ici blize  pocetku
   int j;
   struct oglas_s pomocni;
   for (i = 0; i < n - 1; i = i + 1) {
      ind_raniji = i + 1;
      for (j = i + 2; j < n; j = j + 1) {
         if (oglas[j].cijena > oglas[ind_raniji].cijena ||
             (oglas[j].cijena == oglas[ind_raniji].cijena &&
              oglas[j].povrsina > oglas[ind_raniji].povrsina) ||
             (oglas[j].cijena == oglas[ind_raniji].cijena &&
              oglas[j].povrsina == oglas[ind_raniji].povrsina &&
              oglas[j].sifOglas < oglas[ind_raniji].sifOglas)
            ) {
            ind_raniji = j;
         }
      }
      if (oglas[ind_raniji].cijena > oglas[i].cijena ||
          (oglas[ind_raniji].cijena == oglas[i].cijena &&
           oglas[ind_raniji].povrsina > oglas[i].povrsina) ||
          (oglas[ind_raniji].cijena == oglas[i].cijena &&
            oglas[ind_raniji].povrsina == oglas[i].povrsina &&
            oglas[ind_raniji].sifOglas < oglas[i].sifOglas)
         ) {
         pomocni = oglas[i];
         oglas[i] = oglas[ind_raniji];
         oglas[ind_raniji] = pomocni;
      }
   }

   // ispis
   printf("\n");
   for (i = 0; i < n; i = i + 1) {
      printf("%8d EUR %8d m2 %8d\n", oglas[i].cijena, oglas[i].povrsina, oglas[i].sifOglas);
   }

   return 0;
}
```