

1. Šta radi funkcija **free()** ?

Oslobađa dinamički alociranu memoriju.

2. Ako je data sledeća linija koda: **FILE \*f = fopen("primer.txt", "r");**, kako bi se izvršila provera da li je fajl uspešno otvoren za čitanje?

```
if(f == NULL) printf("Greska prilikom otvaranja fajla.\n");
else printf("Uspesno otvoren fajl.\n");
```

3. Napisati program koji kao argumente prima dve cifre i jedan od operatora +, - ili \* i na osnovu toga pravi odgovarajući izraz i računa njegovu vrednost.

Primer: **./zad 5 4 +**

Izlaz: **5 + 4 = 9**

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    if(argc != 4)
    {
        printf("Greska! Upotreba: %s prva_cifra druga_cifra operator\n(Za operator mnozenja ukucati \\*)\n",argv[0]);
        return 1;
    }
    int a, b, c;
    a = *argv[1] - '0';
    b = *argv[2] - '0';
    switch(*argv[3])
    {
        case '+':
            c = a + b;
            break;
        case '-':
            c = a - b;
            break;
        case '*':
            c = a * b;
            break;
        default:
            printf("Pogresan operator!\n");
            return 1;
    }
    printf("%d %c %d = %d\n", a, *argv[3], b, c);
    return 0;
}
```

```
}
```

4. Ako je dat pokazivač `int *a` napisati jednu liniju koda (jedna instrukcija) koja dinamički alocira niz on 10 celih brojeva pri čemu je vrednost svakog elementa niza automatski nula.

```
a = (int*)calloc(10, sizeof(int));
```

5. Na ulazu je dat niz 21 9 25 20 5 3 14 24 27 7. Koristeći Merge-sort ispisati korake u sređivanju niza.

```
21 9 25 20 5 3 14 24 27 7
9 21 25 20 5 3 14 24 27 7
9 20 21 25 5 3 14 24 27 7
5 9 20 21 25 3 14 24 27 7
3 5 9 14 20 21 24 25 27 7
3 5 7 9 14 20 21 24 25 27
```

6. Šta je rezultat izvršavanja sledećeg koda?

```
#include<stdio.h>

int main()
{
    int x = 5;
    printf("%d", (int)((&x) - (&(x)+1)));
}
-1
```

7. Ako su definisane sledeće promenljive `int a = 5;` `int *p;` `char *c;`, zaokružiti slova ispred komandi koje ne dovode do kompajlerske greške.

- a. `*c = a;`
- b. `c = *a;`
- c. `p = &a;`
- d. `&a = p;`

8. Šta je potrebno da piše umesto **PRVA\_IZMENA** i **DRUGA\_IZMENA** tako da sledeći program štampa niz **a** u rastućem poretku:

```
#include<stdio.h>

int daLiJeVece(int a, int b){ return a > b ? 1 : 0; }

void sortiraj(DRUGA_IZMENA, int a[], int n)
{
    int i, j, t;

    for(i = 0; i < n - 1; i++)
        for(j = i + 1; j < n; j++)
            if(poredjenje(a[i], a[j]))
```

```

        { t = a[i]; a[i] = a[j]; a[j] = t; }
    }

    int main()
    {
        int a[] = {1, 5, 75, 3, 4, 9, 50 }, i;
        sortiraj(PRVA_IZMENA , a, 7);

        for(i = 0; i < 7; i++)
            printf("%d ", a[i]);
    }

```

PRVA\_IZMENA: &daLiJeVece  
 DRUGA\_IZMENA: int (\*poredjenje)(int,int)

9. Šta je rezultat izvršavanja sledećeg koda?

```

#include<stdio.h>

char* funk(char *s, int f)
{
    return f ? ++s : s--;
}

int main()
{
    int i, j;
    char *s = "SPA1";
    s = funk(++s, 0);
    printf("%c\n", *s);
}
P

```

10. Data je sledeća struktura:

```

typedef struct
{
    char *c;           //naziv smestaja
    int brNoc;         //broj nocenja
    float cenaNoc;     //cena po noci
    float ukupnaCena;
} rezervacija;

```

Napisati funkciju **ucitaj** koja za prosleđeni naziv datoteke i niz rezervacija učitava podatke o rezervacijama i smešta ih u niz. Podaci se učitavaju do kraja fajla (maksimalni broj rezervacija je 100)

Format ulaznog fajla:

```

naziv_smestaja_1
br_nocenja_1
cena_nocenja_1
...
naziv_smestaja_i
br_nocenja_i
cena_nocenja_i

```

Po učitavanju jedne rezervacije potrebno je pozvati funkciju **izračunajCenu** koja nema povratnu vrednost i za prosleđenu rezervaciju računa ukupnu cenu po formuli **ukupnaCena = brNoc \* cenaNoc**.

U glavnom delu programa pozvati prethodno definisane funkcije i ispisati za svaki smeštaj naziv i ukupnu cenu.

11. Napisati funkciju **void stampajGlavnu(int \*\*a, int n)** koja štampa glavnu dijagonalu prosleđene matrice čije su dimenzije **nxn bez korišćenja uglastih zagrada za pristup pojedinačnim elementima**.

```

void stampajGlavnu(int **a,int n)
{
    int i;
    for(i=0; i<n; i++) printf("%d\n",*(a + i) + i));
}

```

10.

```

void izracunaCenu(rezervacija *rez)
{
    rez->ukupnaCena = rez->brNoc * rez->cenaNoc;
}
int ucitaj(char *nazivFajla, rezervacija *rez)
{
    FILE *f = fopen(nazivFajla, "r");
    int n = 0, alocirano, duzina;
    char c;
    while(1)
    {
        fscanf(f,"%c",&c);
        if(feof(f)) break;
        alocirano = 0; duzina = 0; rez[n].c = NULL;
        while(c!='\n')
        {
            if(alocirano == duzina)
            {
                alocirano += 5;
                rez[n].c = (char*)realloc(rez[n].c,alocirano*sizeof(char));
            }
            rez[n].c[duzina++] = c;
            fscanf(f,"%c",&c);
        }
    }
}

```

```

    }
    if(alocirano == duzina)
        rez[n].c=(char*)realloc(rez[n].c,(alocirano+1)*sizeof(char));
    rez[n].c[duzina++] = '\0';
    fscanf(f,"%d%f ",&rez[n].brNoc, &rez[n].cenaNoc);
    izracunaCenu(&rez[n]);
    n++;
}
fclose(f);
return n;
}
int main()
{
    char *nazivFajla = "ulaz.txt";
    rezervacija rez[100];
    int n,i;
    n = ucitaj(nazivFajla, rez);
    for(i=0;i<n;i++)
        printf("%s %.2f\n",rez[i].c, rez[i].ukupnaCena);
    return 0;
}

```