



Zadatak 1

- Što će se ispisati sljedećim programskim odsječkom:

```
int a[3], i;  
for (i=0; i<7; i++) a[i/3]=i+2;  
for (i=2; i>=0; i--)  
printf("%d\n", a[i]);
```





Rješenje

```
int a[3], i;  
for (i=0; i<7; i++) a[i/3]=i+2;  
for (i=2; i>=0; i--)  
printf("%d\n", a[i]);
```

Napomena:

```
a[0/3]=a[0]=2;  
a[1/3]=a[0]=3;  
a[2/3]=a[0]=4;  
a[3/3]=a[1]=5;  
a[4/3]=a[1]=6;  
a[5/3]=a[1]=7;  
a[6/3]=a[2]=8;
```

Rješenje:

```
8  
7  
4
```



Zadatak 2

- Učitati veličinu $1 \leq n \leq 10$ jednodimenzionalnog cjelobrojnog polja a
- Učitati elemente polja a
- Pronaći i ispisati najveći broj u polju



Dopisati dijelove naredbi koji nedostaju

```
#include<_____>
#define MAXN 10

int main(){
    int a[MAXN], n, i, max;

    do{
        printf("Unesite velicinu polja: ");
        scanf("%d", &n);
    }
    while (_____);
```



```
for(i=0; _____) {
    printf("\na[%d]= ", i);
    scanf("%d", &a[i]);
}

max=_____
for (i=____; i<n; i++)
    if (a[i]>max) _____;

printf("\nNajveci broj u polju a je %d.\n", _____);

return 0;
}
```




Rješenje

```
#include<stdio.h>
#define MAXN 10

int main(){
    int a[MAXN], n, i, max;

    do{
        printf("Unesite velicinu polja: ");
        scanf("%d", &n);
    }
    while (n<1 || n>10);
```



```
for(i=0; i<n; i++) {
    printf("\na[%d]= ", i);
    scanf("%d", &a[i]);
}

max=a[0];
for (i=1; i<n; i++)
    if (a[i]>max) max=a[i];

printf("\nNajveci broj u polju a je %d.\n", max);

return 0;
}
```



Zadatak 3

- Što će se ispisati sljedećim programom:

```
#include<stdio.h>
#define N 5
int main(){
    char polje[N+1]="ABCDE";
    int i=0;
    while (polje[i]){
        printf("%c%c", polje[i], polje[N-1-i]);
        i++;
    }
    return 0;
}
```





Rješenje

- Što će se ispisati sljedećim programom:

```
#include<stdio.h>
#define N 5
int main(){
    char polje[N+1]="ABCDE";
    int i=0;
    while (polje[i]){
        printf("%c%c", polje[i], polje[N-1-i]);
        i++;
    }
    return 0;
}
```

Rješenje:

AEBDCCDBEA





Zadatak 4

- Napisati program koji učitava niz od 10 znakova i znak *c*.
- Ulazni niz promijeniti na način da se ispred svakog pojavljivanja samoglasnika koji je napisan malim slovom ubaci znak *c*.
- Nakon promjene, niz treba ispisati.
- Zadatak riješiti bez korištenja naredbe *if*.



Nadopunite (1. dio)

```
int main(){
    char c,niz[20+1]; /*treba uzeti dovoljno veliki
                       niz (ekstremni slučaj da su u nizu samo
                       samoglasnici*/
    int i,j,duljina=10;

    printf("Upisite niz od 10 znakova, a zatim (bez
           razmaka) unesite i znak koji treba ubaciti
           ispred svakog samoglasnika\n");

    for(i=0; i<_____ ; _____ )
        scanf("%c",_____);
        scanf("%c", &c);
```



Rješenje (1. dio)

```
int main(){
char c,niz[20+1]; /*treba uzeti dovoljno veliki
niz (ekstremni slučaj da su u nizu samo
samoglasnici*/
int i,j,duljina=10;

printf("Upisite niz od 10 znakova, a zatim (bez
razmaka) unesite i znak koji treba ubaciti
ispred svakog samoglasnika\n");

for(i=0; i<duljina ; i++ )
scanf("%c", &niz[i]);
scanf("%c", &c);
```



Rješenje (2. dio)

```
for(i=0; i<duljina ; i++){
switch(niz[i]){
case 'a':case 'e': case 'i': case 'o': case 'u':
/* pomak svih znakova udesno */
for(j=duljina ; j>i ; j--)
niz[j] = niz[j-1];
niz[i] = c;
duljina++; i++; /* budući da je sad
samoglasnik pomaknut jedno mjestu udesno */
break;
}
}
for(i=0; i<duljina ; i++)
printf("%c", niz[i]);
printf("\n");
return 0;
}
```




Zadatak 5

- Što će se ispisati sljedećim programom:

```
#include<stdio.h>
int main(){
    int f[6]={0,2,3}, i=3;
    do{
        f[i]=f[i-1]+f[i-2];
        printf("%d\n", f[i]);
        i++;
    }while (i<6); return 0;
}
```





Rješenje

- Što će se ispisati sljedećim programom:

```
#include<stdio.h>
int main(){
    int f[6]={0,2,3}, i=3;
    do{
        f[i]=f[i-1]+f[i-2];
        printf("%d\n", f[i]);
        i++;
    }while (i<6);
    return 0;
}
```

Napomena:

```
f[0]=0
f[1]=2
f[2]=3
f[3]=f[2]+f[1]=5
f[4]=f[3]+f[2]=8
f[5]=f[4]+f[3]=13
```

Rješenje:

```
5
8
13
```



Zadatak 6

- Napisati program koji učitava pozitivni cijeli broj koji nema više od 8 znamenki.
 - Ako broj ima 5 znamenki, treba ispisati umnožak znamenki.
 - Ako broj ima 4 znamenke, treba ih redom oduzeti (prva znamenka-druga znamenka- treća znamenka-četvrta znamenka).
 - Ako broj ima 3 znamenke, treba ispisati broj bez središnje znamenke.
 - Ako broj ima 2 znamenke, treba ispisati broj koji čine obrnute znamenke učitano broj.
 - Ako broj ima 1 znamenku, treba ispisati kvadrat te znamenke.
 - Ako broj ima više od 5 znamenki, potrebno je ispisati prve dvije znamenke učitano broj.
- Napomena: zadatak riješiti bez korištenja if naredbe.



Nadopunite

Napisati program koji učitava pozitivni cijeli broj koji nema više od 8 znamenki...

```
#include <stdio.h>
int main()
{
    int broj, znamenke[8]={0}, i=0, brZnam=0, produkt=1;
    do{
        printf("Unesite broj: ");
        scanf("%d", &broj);
    }while ( _____ );
    /* u polju znamenke na mjestu i je znamenka koja je
       na mjestu i u učitanoj broju */

    do{
        znamenke[i]=broj%10;
        broj=broj/10;
        i++;
        brZnam++;
    }while (broj _____ );
```



Rješenje (1. dio)

```
#include <stdio.h>
int main()
{
    int broj, znamenke[8]={0}, i=0, brZnam=0, produkt=1;
    do{
        printf("Unesite broj: ");
        scanf("%d", &broj);
    }while (broj<=0 || broj>999999999);
    /* znamenke[0] je najmanje znacajna znamenka tj.
       prva znamenka s desne strane */

    do{
        znamenke[i]=broj%10;
        broj=broj/10;
        i++;
        brZnam++;
    }while (broj!=0); // ili: while (broj>0)
```

- Ako broj ima 5 znamenki, treba ispisati umnožak znamenki.
- Ako broj ima 4 znamenke, treba ih redom oduzeti (prva znamenka- druga znamenka- treća znamenka- četvrta znamenka).
- Ako broj ima 3 znamenke, treba ispisati broj bez središnje znamenke.

```
switch (brZnam)
{
    case 5:
        for (i=0; ___; ___) produkt*=znamenke[i];
        printf("%d", produkt);
        ___;

    case 4:
        broj = znamenke[3]-znamenke[2]-
        znamenke[1]-znamenke[0];
        printf ("%d", broj);
        ___;

    case 3:
        broj = _____;
        printf ("%d", broj);
        _____;
```




Rješenje (2. dio)

```
switch (brZnam)
{
    case 5:
        for (i=0; i<5; i++) produkt*=znamenke[i];
        printf("%d", produkt);
        break;

    case 4:
        broj = znamenke[3]-znamenke[2]-
              znamenke[1]-znamenke[0];
        printf ("%d", broj);
        break;

    case 3:
        broj = znamenke[2]*10 + znamenke[0];
        printf ("%d", broj);
        break;
```

- 
- Ako broj ima 2 znamenke, treba ispisati broj koji čine obrnute znamenke učitano broj.
 - Ako broj ima 1 znamenku, treba ispisati kvadrat te znamenke.
 - Ako broj ima više od 5 znamenki, potrebno je ispisati prve dvije znamenke učitano broj.

```
case 2:
    broj = _____;
    printf ("%d", broj);
    break;

case 1:
    printf ("%d", znamenke[0]*znamenke[0]);
    break;

default:
    printf ( _____ );

}
return 0;
}
```



Rješenje (3. dio)

```
case 2:
    broj = znamenke[0]*10 + znamenke[1];
    printf ("%d", broj);
    break;

case 1:
    printf ("%d",znamenke[0]*znamenke[0]);
    break;

default:
    printf ("%d %d", znamenke[brZnam-1],
        znamenke[brZnam -2]);

    }
return 0;
}
```