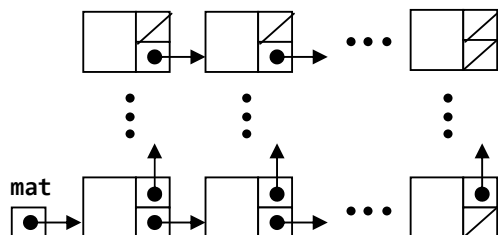


PROGRAMIRANJE II (K2 08.06.2016.)

- ❶ (15 bodova) Pretpostavimo da je cjelobrojna matrica realizovana jednostrukim ulančavanjem elemenata po obje ose (sl. 1). Svaki element matrice ima pokazivače na element iznad i na element desno od sebe. Realizovati funkciju za dodavanje reda na početak postojeće matrice (na dno), sa elementima čije su vrijednosti 0. Prototip funkcije je:

```
void dodajRed(NODE *mat);
```



Sl. 1.

- ❷ (10 bodova) Neka je data biblioteka *stek.h* u kojoj je definisan tip:

```
typedef struct cvor
{
    char z;
    struct cvor *sljedeci;
} CVOR;
```

i funkcije:

```
void push(CVOR **tos, char z);
int pop(CVOR **tos, char *z);
```

Definisati funkciju koja, pomoću steka, provjerava i vraća informaciju o tome da li su pravilno uparene zagrade u zadatom stringu. Prototip funkcije je:

```
int zagrade(const char *str);
```

Primjeri:

```
zagrade("{a*(b-10)+1}*2") → 1
zagrade("(a*{b-10}+1)*2") → 1
zagrade("[a*(b-10)+1*2") → 0
zagrade("{a*(b-10}+1)*2") → 0
```

- ❸ (10 bodova) Definirati funkciju koja određuje i kao rezultat vraća dubinu binarnog stabla. Čvorovi stabla definisani su sljedećim tipom:

```
typedef struct node
{
    void *info;
    struct node *left, *right;
} NODE;
```

Prototip funkcije je:

```
int dubina(NODE *root);
```

- ❹ (15 bodova) Neka je definisan tip:

```
typedef struct graf
{
    int n; // broj cvorova
    void *info[MAX]; // inf.sadrzaj cvorova
    int ms[MAX][MAX]; // matrica susjednosti
} GRAF;
```

kojim se reprezentuje neusmjeren graf. Definirati funkciju koja provjerava i vraća informaciju o tome da li je zadati graf povezan, a čiji je prototip:

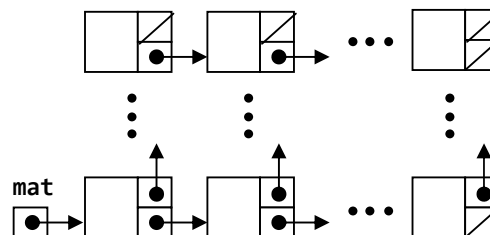
```
int povezan(GRAF *g);
```

Napomena: Graf je povezan ako između svaka dva čvora postoji putanja.

PROGRAMIRANJE II (K2 08.06.2016.)

- ❶ (15 bodova) Pretpostavimo da je cjelobrojna matrica realizovana jednostrukim ulančavanjem elemenata po obje ose (sl. 1). Svaki element matrice ima pokazivače na element iznad i na element desno od sebe. Realizovati funkciju za dodavanje reda na početak postojeće matrice (na dno), sa elementima čije su vrijednosti 0. Prototip funkcije je:

```
void dodajRed(NODE *mat);
```



Sl. 1.

- ❷ (10 bodova) Neka je data biblioteka *stek.h* u kojoj je definisan tip:

```
typedef struct cvor
{
    char z;
    struct cvor *sljedeci;
} CVOR;
```

i funkcije:

```
void push(CVOR **tos, char z);
int pop(CVOR **tos, char *z);
```

Definisati funkciju koja, pomoću steka, provjerava i vraća informaciju o tome da li su pravilno uparene zagrade u zadatom stringu. Prototip funkcije je:

```
int zagrade(const char *str);
```

Primjeri:

```
zagrade("{a*(b-10)+1}*2") → 1
zagrade("(a*{b-10}+1)*2") → 1
zagrade("[a*(b-10)+1*2") → 0
zagrade("{a*(b-10}+1)*2") → 0
```

- ❸ (10 bodova) Definirati funkciju koja određuje i kao rezultat vraća dubinu binarnog stabla. Čvorovi stabla definisani su sljedećim tipom:

```
typedef struct node
{
    void *info;
    struct node *left, *right;
} NODE;
```

Prototip funkcije je:

```
int dubina(NODE *root);
```

- ❹ (15 bodova) Neka je definisan tip:

```
typedef struct graf
{
    int n; // broj cvorova
    void *info[MAX]; // inf.sadrzaj cvorova
    int ms[MAX][MAX]; // matrica susjednosti
} GRAF;
```

kojim se reprezentuje neusmjeren graf. Definirati funkciju koja provjerava i vraća informaciju o tome da li je zadati graf povezan, a čiji je prototip:

```
int povezan(GRAF *g);
```

Napomena: Graf je povezan ako između svaka dva čvora postoji putanja.