JMBAG **IME I PREZIME**

Algoritmi i strukture podataka – 1. Ispitni rok

2. srpnja 2013.

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe ***goto****. Neefikasna rješenja mogu donijeti manje bodova. Nerekurzivne funkcije se ne priznaje kao rješenja u zadatcima u kojima se traži rekurzivna funkcija. Ispit nosi maksimalno 70 bodova, a prag za prolaz pismenog ispita je* ***35*** *bodova. 5. zadatak rješavate na ovom obrascu dok ostale zadatke rješavate na svojim listovima papira. Ovaj obrazac morate predati.*

Zadatak 1. (14 bodova)

Napišite funkciju koja će u ulazno polje cijelih jednoznamenkastih brojeva dodati broj 0 nakon svakog pojavljivanja broja 5, a u skladu s tim i dinamički povećati veličinu polja na osnovu novog broja članova. Funkcija kao rezultat vraća pokazivač na navedeno polje. Prototip funkcije je:

```
int *dodaj_nula_iza_pet(int *polje, int *br_clanova)
```

Također, napisati i glavni program u kojem će se učitati inicijalna veličina polja, polje dinamički alocirati, napuniti polje do maksimalnog kapaciteta slučajno odabranim vrijednostima, pozvati funkcija i ispisati novo polje te osloboditi dinamički stvorenu memoriju.

Primjer: [9 4 2 5 1 5 3] -> [9 4 2 5 0 1 5 0 3]

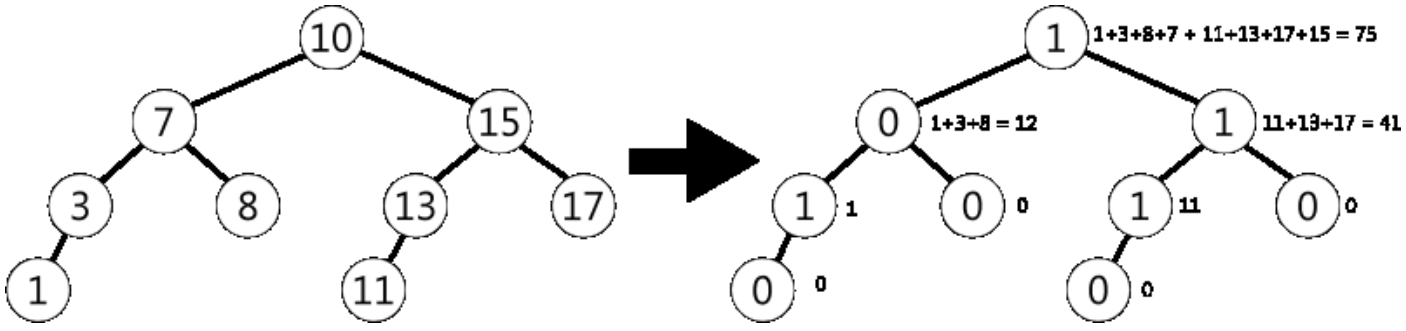
Zadatak 2. (14 bodova)

Binarno stablo sadrži cjelobrojne elemente, a čvor je definiran sljedećom strukturom:

```
typedef struct s {
      int vrijednost;
      struct s *lijevo, *desno;
} cvor;
```

Napisati funkciju koja će u svakom čvoru zamijeniti vrijednost sa 0 ako je zbroj svih potomaka tog čvora paran odnosno sa 1 ako je zbroj svih potomaka neparan.

Primjer očekivane transformacije:



Zadatak 3. (14 bodova)

Zadan je niz brojeva: **17, 24, 33, 19, 78, 49, 67, 28**.

a) **(7 bodova)** Ilustrirajte (nacrtajte stablo nakon svake promjene) stvaranje gomile s relacijom ***veći od*** (max heap) od zadanog polja brojeva algoritmom čija je složenost za najgori slučaj *O(n).*

b) **(7 bodova)** Za gomilu iz a) zadatka prikažite postupak uzlaznog *heapsorta*. Prikažite svaki korak sortiranja (nacrtajte stablo nakon svake zamjene dva elementa).

Zadatak 4. (14 bodova)

Za tip podatka `Stog` koji je realiziran jednostruko povezanom listom definirane su funkcije za inicijalizaciju stoga, dodavanje elementa na stog te skidanje elementa sa stoga. Elementi stoga su nizovi znakova koji sadrže po jedan operator ("+" ili "*") ili cijeli broj zapisan kao niz znakova. Prototipovi navedenih funkcija su:

```
void init_stog(Stog *stog);
int dodaj(char *element, Stog *stog);
int skini(char *element, Stog *stog);
```

Funkcije `dodaj` i `skini` vraćaju 1 ako je operacija dodavanja ili skidanja uspjela, a 0 inače.

Napisati **rekurzivnu** funkciju `void izracunaj(Stog *stog)` koja će izračunati matematičke izraze pohranjene na stogu u notaciji u kojoj svaki operator slijedi nakon svih svojih operanada. Pretpostavlja se da su operatori binarni, dakle djeluju na po dva operanda (cijeli broj ili izraz kojemu je brojčana vrijednost cijeli broj).

Rezultat izraza nakon izvršavanja funkcije treba ostati na stogu, a sve ostalo tijekom izračunavanja (uključujući sve operatore i operande) treba treba biti uklonjeno.

Za pretvorbu cijelog broja (int) u niz znakova koristiti funkciju čiji je prototip:

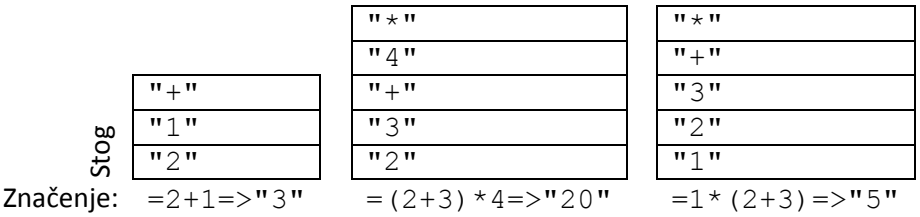
```
char *intustr(int val);
```

Za pretvorbu niza znakova u cijeli broj (int) koristiti funkciju čiji je prototip:

```
int struint(char *str);
```

Funkcije za pretvorbu nije potrebno pisati.

Primjeri stogova sa značenjem izraza:



Napomene: Rješenja koja neće koristiti zadane funkcije za rad sa stogom donijet će manje bodova.

Zadatak 5. (14 bodova)

Prikažite sadržaj stoga u trenutku neposredno prije prvog izvođenja naredbe **return 0** u funkciji **f2** i naznačite broj bajtova koje zauzima svaka od varijabli.

```
int f2(double *polje2, int n2, int d) {
	int i, brojac=0;
	if (n2<=0){
		return 0;
	}
	for(i=1;i<=n2/d;i++){
		brojac+=f2(polje2+i+d,n2-i-d, d);
	}
	return brojac+1;
}
void f1(double *polje1, int n1){
	f2(polje1,n1, 3);
}
int main() {
	double polje[8];
	...
	f1(polje, 8);
}
```

Rješenja:

Zadatak 1. (14 bodova)

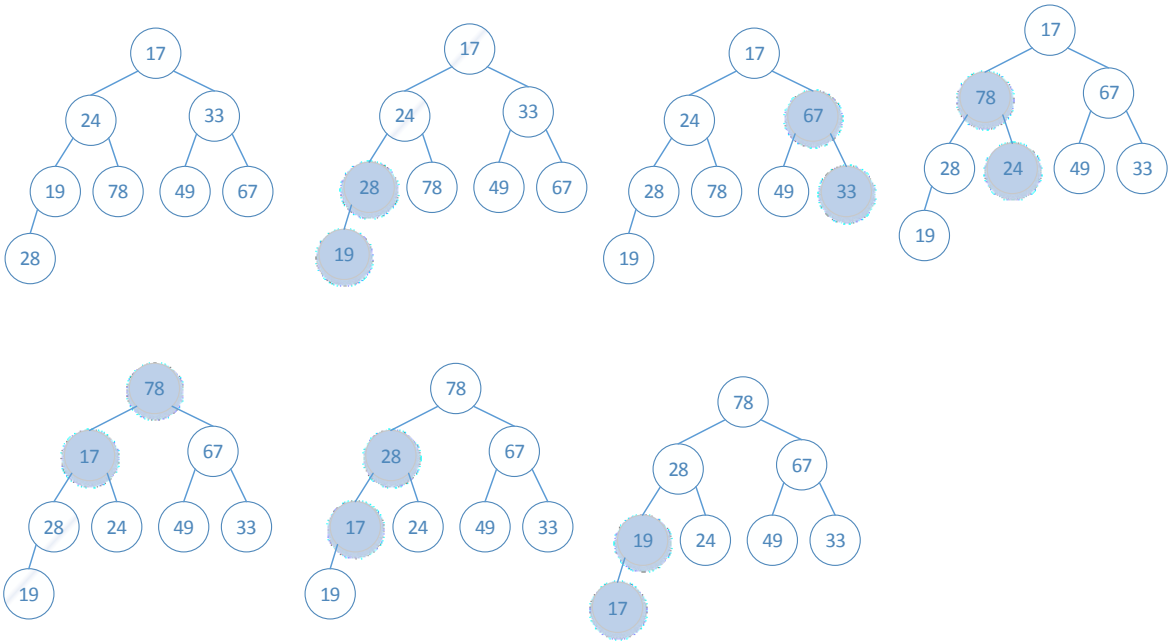
```
int *dodaj_nula_iza_pet(int *polje, int *br_clanova){
	int i, j;
	for(i=0 ; i < *br_clanova ; i++){
		if (polje[i] == 5){
			(*br_clanova)++;
			polje = (int*) realloc(polje, (*br_clanova) *sizeof(int));
			for (j = *br_clanova - 1; j > i + 1; j--) {
				polje[j] = polje[j-1];
			}
			polje[j] = 0;
			i++; // može, ali i ne mora biti
		}
	}
	return polje;
}

int main(){
	int *polje, i, br_clanova;
	scanf("%d\n", &br_clanova)
	polje = (int*) malloc(br_clanova * sizeof(int));
	srand((unsigned) time(NULL));
	for(i=0; i<br_clanova ; i++){
		polje[i] = rand()%10;
	}
	printf("\nPrije dodavanja nula\n");
	for(i=0; i<br_clanova ; i++){
		printf("%d\n", polje[i]);
	}
	polje = dodaj_nula_iza_pet(polje, &br_clanova);
	printf("\nNakon dodavanja nula\n");
	for(i=0; i<br_clanova ; i++){
		printf("%d\n", polje[i]);
	}
	free(polje);
	return 0;
}
```

Zadatak 2. (14 bodova)

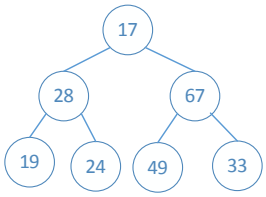
```
int fun(cvor *korijen){
  int suma;
  int trenutnaVrijednost;

  if(korijen == NULL){
    return 0;
  }

  suma = fun(korijen->lijevo) + fun(korijen->desno);
  trenutnaVrijednost = korijen->vrijednost;
  korijen->vrijednost = suma % 2;
  return suma + trenutnaVrijednost;
}
```

Zadatak 3. (14 bodova)

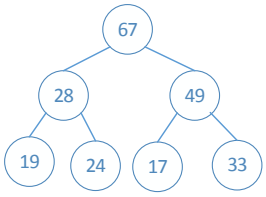
a)



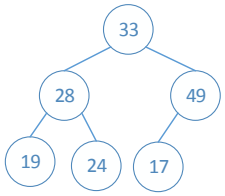
b)



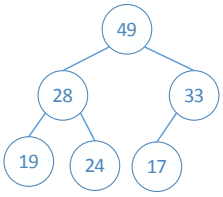
17, 28, 67, 19, 24, 49, 33, 78



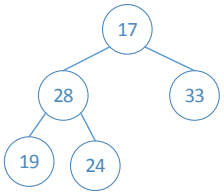
67, 28, 49, 19, 24, 17, 33, 78



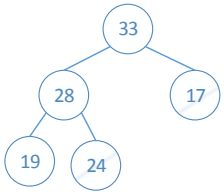
33, 28, 49, 19, 24, 17, 67, 78



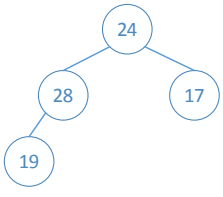
49, 28, 33, 19, 24, 17, 67, 78



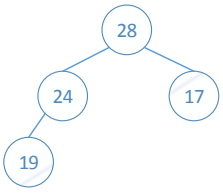
17, 28, 33, 19, 24, 49, 67, 78



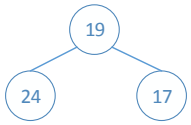
33, 28, 17, 19, 24, 49, 67, 78



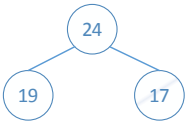
24, 28, 17, 19, 33, 49, 67, 78



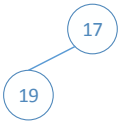
28, 24, 17, 19, 33, 49, 67, 78



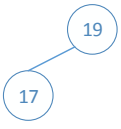
19, 24, 17, 28, 33, 49, 67, 78



24, 19, 17, 28, 33, 49, 67, 78



17, 19, 24, 28, 33, 49, 67, 78



19, 17, 24, 28, 33, 49, 67, 78

17, 19, 24, 28, 33, 49, 67, 78

Zadatak 4. (14 bodova)

```
void izracunaj(Stog *stog){
          char e[10], e1[10]={0}, e2[10]={0};
          int rez;
          if(skini(e, stog)){
                    izracunaj(stog);
                    if(isdigit(e[0])) dodaj(e,stog);
                    else{
                              skini(e1, stog);
                              skini(e2, stog);
                              if(e[0]=='+') rez = struint(e1)+struint(e2);
                              else rez = struint(e1)*struint(e2);
                              dodaj(intustr(rez), stog);
                    }
          }
}
```

Zadatak 5. (14bodova)

brojac	4
i	4
Povr. f2	4
polje2	4
n2=0	4
d=3	4
brojac	4
i	4
Povr. f2	4
polje2	4
n2=4	4
d=3	4
brojac	4
i	4
Povr. f2	4
polje2	4
n2=8	4
d=3	4
Povr. f1	4
polje1	4
n1=8	4