

1	2	3	4	5	6	Σ

BROJ INDEKSA

IME I PREZIME

PREDAVAČ

ASISTENT

Programiranje (C) - 3. kolokvij, 7.6.2002.

Uputa: Zabranjeno je korištenje bilo kakvih pomagala. Rješenja pišite desno ili ispod zadatka. Predajte samo ovaj list.

1. (12 bod.) a) Ako su dane sljedeće deklaracije:

```
char rijec[]="Adam";
```

```
char *prijec=rijec;
```

kakav ispis daju naredbe:

```
i) printf("%s", prijec+2);
```

Rj: am

```
ii) *(prijec)='d'; printf("%s", prijec);
```

Rj: ddam

b) i) Opišite smisao sljedeće deklaracije: `char *(*x)(int *a[])`

Rj: Deklariran je pokazivač na funkciju koja vraća pokazivač na tip char, a kao argument prima polje pokazivača na tip int.

ii) Deklarirajte pokazivač na funkciju čiji je ulazni argument polje pokazivača na tip int, a vraća pokazivač na tip znak

Rj: to je upravo ovo pod i.

2. (8 bod.) Nađite, opišite i ispravite greške u sljedećem programu:

```
struct Datum{
    int dan, mjesec, godina;
};
struct Racun{
    char ime [30];
    int broj;
    struct Datum datum_otvaranja;
    float stanje;
};
int main()
{
    struct Datum danas={17,3,1999}, sutra;
    struct Racun pero;
    Racun.Datum.dan=12;
    \* nastavak programa ....*\
    return 0;
}
```

Rj: Ne može biti *Racun.Datum.dan* jer su to imena struktura, već moraju biti varijable, npr. *pero.datum_otvaranja.dan*

3. (20 bod.) Zadana je struktura

```
struct kazaliste {
    char predstava[50], redatelj[30];
    int br_posjetilaca;
};
```

Napišite program koji učitava najviše 20 takvih struktura i zatim ispisuje ime najposjećenije predstave i ime njezinog redatelja. Rj:

```
#include<stdio.h>
#include<stdlib.h>

struct kazaliste {
    char predstava[50], redatelj[30];
    int br_posjetilaca;
};
main(){
    struct kazaliste r[20];
    int i,n, ind, max=0;
    printf("Koliko predstava zelite unijeti (max 20)?");
    scanf("%d",&n);
    while( ! (n>0 && n<=20) ){
        printf("Greska, broj mora biti veci od 0, a manji od 20 \n");
        scanf("%d",&n);
    }
    for (i=0 ; i<n ; i++){
        printf("Predstava broj %d\n", i+1);
        printf("Naziv:");
        scanf("%s", r[i].predstava);
        printf("Redatelj:");
        scanf("%s", r[i].redatelj);
        printf("Broj posjetilaca:");
        scanf("%d", &r[i].br_posjetilaca);
    }
    for (i=0 ; i<n ; i++){
        if (r[i].br_posjetilaca > max){
            max = r[i].br_posjetilaca;
            ind = i;
        }
    }
    printf("Predstava %s (redatelj: %s) je najposjecenija sa %d gledatelja\n",
        r[ind].predstava, r[ind].redatelj, r[ind].br_posjetilaca);
    return 0;
}
```

4. (20 bod.) a) Napišite što donji program ispisuje. Obrazložite vaš odgovor.

```
#include <stdio.h>
int a,b,c;
void zbuni1 (int b, int &a){
    a+=3; b+=2; c+=1;
}
void zbuni2 (int b, int *a){
    *a+=1; b+=2; c+=3;
}
void zbuni3 (int *b, int c){
    a+=1; *b+=2; c+=3;
}
int main(){
    int a,b,c; a=b=c=0;
    zbuni1(c,&b); zbuni2(b,&c); zbuni3(&a,a);
    printf("%d %d %d",a,b,c);
    return 0;
}
```

Rj: Ispisuje: 2 3 1. *zbuni1* promijeni vrijdnost od b jer se prenosi po adresi (poveća mu se vrijednost za 3) ...

5. (20 bod.) Napišite funkciju koja prima pokazivač na početak niza, duljinu niza te vraća aritmetičku i geometrijsku sredinu članova niza. Također napišite program koji kreira takav jedan niz (duljina niza je proizvoljna, treba je učitati sa tipkovnice), poziva gore navedenu funkciju i ispisuje tražene rezultate. (Napomena : Ne zaboravite deklarirati sve potrebne varijable)

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

void f(int *niz, int n, double *as, double *gs){
    int i;
    if (n<1) return;
    *as = 0; *gs = 1;
    for (i=0; i<n ; i++){
        *as += niz[i];
        *gs *= niz[i];
    }
    *as = *as / n;
    *gs = pow(*gs, 1.0/n);
} main(){
    int i,n, *niz;
    double as, gs;
    scanf("%d", &n);
    if (n>1){
        niz = (int*) malloc(n*sizeof(int));
        for (i=0;i<n;i++){
            scanf("%d", niz+i);
        }
        f(niz,n,&as, &gs);
        printf("AS=%lf , GS=%lf\n", as,gs);
    }
}
```

6. (25 bod.) Napišite funkciju koja prima pokazivač na vezanu listu cijelih brojeva i iz liste izbacuje sve neparne brojeve. Funkcija treba vratiti pokazivač na početak modificirane liste.

```
struct brojevi* preuredi(struct brojevi* pocetak){
    struct brojevi *trenutni, *prethodni;

    while(pocetak!=NULL && pocetak->broj % 2){
        struct brojevi *temp = pocetak;
        pocetak = pocetak->next;
        free(temp);
    }
    if (pocetak==NULL) return NULL;
    prethodni = pocetak;
    trenutni = prethodni->next;

    while(trenutni!=NULL){
        if (trenutni->broj % 2){
            /*izbacujem ga iz liste*/
            prethodni->next = trenutni->next;
            free(trenutni);
            trenutni = prethodni->next;
        }
        else{
            prethodni = trenutni;
            trenutni = trenutni->next;
        }
    }
    return pocetak;
}
```

1	2	3	4	5	6	Σ

BROJ INDEKSA

IME I PREZIME

PREDAVAČ

ASISTENT

Programiranje (C) - 3. kolokvij, 7.6.2002.

Uputa: Zabranjeno je korištenje bilo kakvih pomagala. Rješenja pišite desno ili ispod zadatka. Predajte samo ovaj list.

1. (12 bod.) a) Ako su dane sljedeće deklaracije:

```
char rijec[]="Eva";
```

```
char *prijec=rijec;
```

kakav ispis daju naredbe:

```
i) *(prijec+2)='o'; printf("%s", prijec);
```

Rj: Evo

```
ii) *prijec++='a'; printf("%s", prijec);
```

Rj: va

b) i) Opišite smisao sljedeće deklaracije: `double *(*x[20])(int *)`

Rj: Deklarira polje pokazivača veličine 20 na funkciju koja vraća pokazivač na tip double, a kao argument uzima pokazivač na tip int.

ii) Deklarirajte polje od 12 elemenata koji su pokazivači na funkcije. Svaka funkcija kao ulazne argumente ima dva pokazivača na tip double, a vraća pokazivač na tip float

Rj: `char* (*f)(int *x[])`

2. (8 bod.) Nadite, opišite i ispravite greške u sljedećem programu:

```
struct zgrada {
    int tip;
    char ime[30];
    char adresa [40];
    int br_soba;
} hotel;
int main()
{
    zgrada.tip = 3;
    printf("%d\n", hotel->tip);
    \* nastavak programa ....*\
    return 0;
}
```

Rj: ne može biti `hotel` → `tip` jer `hotel` nije pokazivač, već mora biti `hotel.tip`. Također ne može biti `zgrada.tip` već mora biti `neka_arijabla_tipa_zgrada.tip` (npr. `hotel.tip`)

3. (20 bod.) Zadana je struktura

```
struct radnik{
    char ime[20], prezime[30];
    int staz;
    float placa;
};
```

Napišite program koji učitava najviše 20 takvih struktura i zatim računa i ispisuje najveću plaću te prosječnu plaću svih radnika.

Rj:

```
#include<stdio.h>
#include<stdlib.h>

struct radnik{
    char ime[20],prezime[30];
    int staz;
    float placa;
};

main(){
    struct radnik r[20];
    int i,n, ind; float max=0.0f,avg = 0.0f;
    printf("Koliko radnika zelite unijeti (max 20)?");
    scanf("%d",&n);
    while( ! (n>0 && n<=20) ){
        printf("Greska, broj mora biti veci od 0, a manji od 20 \n");
        scanf("%d",&n);
    }
    for (i=0 ; i<n ; i++){
        printf("Radnik broj %d\n", i+1);
        printf("Ime:");
        scanf("%s", r[i].ime);
        printf("Prezime:");
        scanf("%s", r[i].prezime);
        printf("Staz:");
        scanf("%d", &r[i].staz);
        printf("Placa:");
        scanf("%f", &r[i].placa);
    }
    for (i=0 ; i<n ; i++){
        if (r[i].placa > max){
            max = r[i].placa;
            ind = i;
        }
        avg += r[i].placa;
    }
    printf("Radnik %s %s ima najvecu placu %f\n",r[ind].ime, r[ind].prezime, r[ind].placa);
    printf("Prosjecna placa= %f\n",avg/n);
    return 0;
}
```

4. (20 bod.) a) Napišite što donji program ispisuje. Obrazložite vaš odgovor.

```
#include <stdio.h>
int a,b,c;
void zbuni1 (int b, int a){
    a+=1; b+=1; c+=1;
}
void zbuni2 (int b, int *a){
    *a+=2; b+=2; c+=2;
}
void zbuni3 (int *b, int c){
    a+=3; *b+=3; c+=3;
}

int main(){
    int a,b,c; a=b=c=0;
    zbuni1(a,b); zbuni2(b,&c); zbuni3(&a,c);
    printf("%d %d %d",a,b,c);
    return 0;
}
```

Rj: Ispisuje 3 0 2 jer se kod *zbuni2* *c* prenosi po adresi i poveća se za 2 i kod *zbuni3* *a* prenosi po adresi i poveća se za 3. Globalne varijable su mijenjale svoje vrijednosti, ali one nas trenutno ne zanimaju, već samo one deklarirane u main funkciji.

5. (20 bod.) Napišite funkciju koja prima pokazivač na početak niza, duljinu niza te vraća njegov najveći i najmanji element. Također napišite program koji kreira takav jedan niz (duljina niza je proizvoljna, treba je učitati sa tipkovnice), poziva gore navedenu funkciju i ispisuje tražene rezultate. (Napomena : Ne zaboravite deklarirati sve potrebne varijable)

```
#include<stdio.h>
#include<stdlib.h>

void f(int *niz, int n, int *max, int *min){
    int i;
    if (n<1) return;
    *max = *min = niz[0];
    for (i=1; i<n ; i++){
        if (niz[i] < *min) *min = niz[i];
        else if (niz[i] > *max) *max = niz[i];
    }
}

main(){
    int i,n, max, min, *niz;
    scanf("%d", &n);
    if (n>1){
        niz = (int*) malloc(n*sizeof(int));
        for (i=0;i<n;i++){
            scanf("%d", niz+i);
        }
        f(niz,n,&max, &min);
        printf("max=%d , min=%d\n", max,min);
    }
}
```

6. (25 bod.) Napišite funkciju koja prima pokazivač na vezanu listu cijelih brojeva i mijenja tu vezanu listu na način da se na početku liste trebaju naći neparni, a zatim iza njih parni brojevi. Funkcija treba vratiti pokazivač na početak tako modificirane liste. Redoslijed parnih i neparnih brojeva ne mora ostati isti. Ne smijete koristiti pomoćnu vezanu listu.

```
struct brojevi* preuredi(struct brojevi* pocetak){
    /*ideja: kad naidjem na neparni stavim ga na pocetak liste*/
    struct brojevi *trenutni = pocetak, *prethodni;
    while(trenutni!= NULL && trenutni->broj % 2)
        trenutni = trenutni->next;

    if (trenutni==NULL) return pocetak;

    /*u ovom trenutku sam naisao na parni broj */
    prethodni = trenutni;
    trenutni = prethodni->next;

    while(trenutni!=NULL){
        if (trenutni->broj % 2){
            prethodni->next = trenutni->next;

            /*stavi ga na pocetak liste*/
            trenutni->next = pocetak;
            pocetak = trenutni;

            trenutni = prethodni->next;
        }
        else{
            prethodni = trenutni;
            trenutni = trenutni->next;
        }
    }
    return pocetak;
}
```