

FER2 – ASP – 2006/2007 – GRUPA 9
VJEŽBANJE ZA 2. BLIC

SORT-01

A posteriori analiza pokazala je da na nekom računalu sortiranje polja od 10000 elemenata algoritmom Selection Sort traje otprilike 1 sekundu. Kolika je realna procjena za vrijeme sortiranja dvostruko duljeg polja od 20000 elemenata na istom računalu istim algoritmom?

- a) Više od 1 sekunde, no manje od 2 sekunde.
 - b) Otprilike 2 sekunde.
 - c) Više od 2 sekunde no manje od 4 sekunde.
 - d) Otprilike 4 sekunde.
 - e) Više od 4 sekunde no manje od 8 sekundi.
-

SORT-02

A posteriori analiza pokazala je da na nekom računalu sortiranje polja od 100000 elemenata algoritmom Merge Sort traje u prosjeku 1 sekundu. Kolika je realna procjena za vrijeme sortiranja dvostruko duljeg polja od 200000 elemenata na istom računalu istim algoritmom?

- a) Otprilike 1 sekundu.
 - b) Više od 1 sekunde no znatno manje od 2 sekunde.
 - c) Otprilike 2 sekunde.
 - d) Više od 2 sekunde no znatno manje od 4 sekunde.
 - e) Barem 4 sekunde.
-

SORT-03

Apriorne ocjene za vrijeme sortiranja polja od n elemenata algoritmom Quick Sort glase:

- a) $\Theta(n \log n)$ u prosječnom i u najgorem slučaju.
 - b) $\Theta(n \log n)$ u prosječnom slučaju, $\Theta(n^2)$ u najgorem slučaju.
 - c) $\Theta(n^2)$ u prosječnom i u najgorem slučaju.
 - d) $\Theta(n \log n)$ u prosječnom slučaju, $\Theta(n^{3/2})$ u najgorem slučaju.
 - e) $\Theta(n^{3/2})$ u prosječnom i u najgorem slučaju.
-

SORT-04

Polje:

17	31	45	43	11	24	8
----	----	----	----	----	----	---

 uzlazno sortiramo algoritmom Quick Sort. Stožer se bira kao medijan između tri elementa koji se nalaze na početku, kraju odnosno sredini polja. Odredite medijan koji će biti izabran u prvom koraku algoritma.

- a) 8
 - b) 24
 - c) 17
 - d) 43
 - e) ni jedan od ponuđenih
-

SORT-05

Polje [17 31 3 43 11 24 8] uzlazno sortiramo algoritmom Quick Sort. Stožer je početni element polja. Odredite dva pod-polja koja će nakon prvog koraka algoritma biti podvrgnuta rekurzivnim pozivima istog algoritma.

- a) Prazno polje , [31 43 11 24 8]
 - b) [31 3 43] , [11 24 8]
 - c) [3 11 8] , [31 43 24]
 - d) [3 8 11] , [24 31 43]
 - e) [8 3 11] , [43 24 31]
-

SORT-06

Polje [17 31 3 43 11 24 8] uzlazno sortiramo algoritmom Shell Sort, uz primjenu inkrementalnog slijeda brojeva: $3 > 2 > 1$. Odredite izgled polja nakon prve faze Shell Sort-a, dakle nakon provedenog 3-subsorta.

- a) [8 11 3 17 31 24 43]
 - b) [3 8 11 17 24 31 43]
 - c) [11 24 3 43 17 31 8]
 - d) [3 24 8 31 11 43 17]
 - e) [8 31 3 17 11 24 43]
-

OO-01

Zadan je program u jeziku C++:

```
class MojRazred {
    int _MojPodatak;
};
int main ( ) {
    MojRazred obj;
    obj._MojPodatak = 10;
    return 0;
}
```

Koja od sljedećih tvrdnji je točna?

- a) Program se može prevesti, povezati i izvoditi bez greške.
 - b) Program se prevodi uz upozorenje (warning), no ipak se može povezati i izvoditi bez greške.
 - c) Program se može prevesti i povezati. Kod izvođenja, naredba za pridruživanje se ignorira, no to ne izaziva grešku.
 - d) Program se može prevesti i povezati. Naredba za pridruživanje se izvršava, no nakon nje se «baca» iznimka.
 - e) Program se ne može prevesti jer prevodilac javlja grešku.
-

OO-02

Zadan je razred u jeziku C++:

```
class MojRazred {
    int _MojPodatak;
};
```

U glavnoj funkciji deklariran je objekt:

MojRazred obj;

Kako izgleda naredba kojom glavna funkcije članskoj varijabli _MojPodatak unutar objekta obj pridružuje vrijednost 20?

- a) obj._MojPodatak = 20;
 - b) obj -> _MojPodatak = 20;
 - c) MojRazred *pobj = &obj; (*pobj)._MojPodatak = 20;
 - d) Takva naredba postoji, no nije ni jedna od gore navedenih.
 - e) Takva naredba ne postoji.
-

OO-03

Promatramo dvije deklaracije u jeziku C++:

```
struct MyStruct {                class MyClass {
    int _Mydata;                  int _MyData;
};                                };

```

Koja od sljedećih tvrdnji je točna?

- a) Dvije deklaracije su ekvivalentne u smislu da odgovarajući objekti imaju ista svojstva.
 - b) Članska varijabla u MyStruct je javna a članska varijabla u MyClass je privatna.
 - c) Prva deklaracija nije dozvoljena jer C++ ne poznaje struct.
 - d) Druga deklaracija nije dozvoljena jer razred mora imati barem jednu člansku funkciju.
 - e) Obje deklaracije su sintaktički neispravne, zato jer su im članske varijable privatne, a ne postoje članske funkcije koje bi mijenjale te varijable.
-

OO-04

Promatramo sljedeći C++ program:

```
#include <stdio.h>

class MojRazred {
public:
    MojRazred( );
    int * _Podaci;
};
MojRazred :: MojRazred( ) {
    _Podaci = new int [10];
    for ( int i=0; i<10; i++ )
        _Podaci[i] = 0;
}
void Promijeni (MojRazred obj) {
    obj._Podaci[0] = 1;
}
int main ( ) {
    MojRazred ob;
    Promijeni(ob);
    printf("%d \n", ob._Podaci[0]);
    return 0;
}

```

Odaberite točnu tvrdnju:

- a) Program ispisuje 0 .
 - b) Program ispisuje 1 .
 - c) Program ispisuje cjelobrojnu vrijednost koja nije definirana.
 - d) Program se ne može prevesti zato jer u razredu MojRazred nije definiran copy konstruktor.
 - e) Program neće ništa ispisati zato jer će doživjeti grešku tijekom izvođenja pri pokušaju mijenjanja članske varijable unutar objekta.
-

OO-05

U C++ programu nalazi se sljedeća deklaracija:

```
class MyClass {  
    public:  
        MyFunc( );  
    private:  
        int _MyData;  
};
```

U glavnoj funkciji stvoren je objekt iz razreda MyClass na sljedeći način:

```
MyClass *p = new MyClass( );
```

Koja od sljedećih naredbi u nastavku glavne funkcije predstavlja ispravan poziv članske funkcije MyFunc() ?

- a) p-> MyFunc();
 - b) p.MyFunc();
 - c) MyFunc(*p);
 - d) MyFunc(p);
 - e) (*p)->MyFunc();
-

OO-06

Zadan je sljedeći C++ program:

```
#include <stdio.h>  
  
class MyClass {  
    public:  
        MyClass( );  
        MyClass(int a);  
        ~MyClass( );  
    private:  
        int _data;  
};  
MyClass :: MyClass ( ) {  
    _data = 0;  
    printf("%d ", _data);  
}  
MyClass :: MyClass (int a) {  
    _data = a;  
    printf("%d ", _data);  
}  
MyClass :: ~MyClass ( ) {  
    printf("destroy %d ", _data);  
}  
int main ( ) {  
    MyClass x;  
    MyClass *y = new MyClass(5);  
    delete y;  
}
```

Što će ispisati ovaj program?

- a) 0 5 destroy 5 destroy 0
 - b) 0 5 destroy 0 destroy 5
 - c) 0 5 destroy 5
 - d) 5 destroy 5
 - e) 0 0 destroy 0 destroy 0
-

STOG-01

Koja od sljedećih nizova naredbi u pseudokodu će zamijeniti vrijednost varijabli A i B pomoću stoga:

- a) stavi(A); stavi(B); skini(A); skini(B);
 - b) stavi(A); skini(B);
 - c) stavi(B); skini(A);
 - d) stavi(A); stavi(Pom); stavi(B); stavi(Pom); skini(A);
stavi(Pom);skini(B);
 - e) stavi(A); skini(B); stavi(B); skini(A);
-

STOG-02

Na stog prikazan poljem pohranjuju se samo cijeli brojevi. Prototip funkcije za skidanje cijelog broja sa stoga je (funkcija vraća 0 ili 1, ovisno o tome da li se zapis uspio skinuti s vrha stoga):

- a) int skini(int stavka, int stog[], int *vrhStog);
 - b) int skini(int *stavka, int stog[], int *vrhStog);
 - c) int skini(float stavka, float stog[], int vrhStog);
 - d) int *skini(int *stavka, int stog[], int vrhStog);
 - e) void *skini(int *stavka, int stog[], int n, int *vrhStog);
-

STOG-03

Složenost funkcije

```
int dodaj (zapis stavka, zapis stog[], int n, int *vrh) {  
    if (*vrh >= n-1) return 0;  
    (*vrh)++;  
    stog [*vrh] = stavka;  
    return 1;  
}
```

je:

- a) $O(n)$
 - b) $O(\log n)$
 - c) složenost ovisi o veličini zapisa stavke, pa se ne može jednoznačno odrediti
 - d) $O(1)$
 - e) $O(\log_2 n)$
-

STOG-04

Ako je stog realiziran cjelobrojnim poljem od n elemenata, kolika je apriorna složenost skidanja SVIH elemenata sa stoga:

- a) $O(1)$
 - b) $O(n)$
 - c) $O(n^2)$
 - d) $O(\log_2 n)$
 - e) ovisi o operacijskom sustavu
-

STOG-05

Ako funkcija stavljanja na stog vraća 1 u slučaju uspjeha a 0 u slučaju neuspjeha i ima prototip

```
int push (int element);
```

a funkcija skidanja sa stoga vraća vrijednost elementa s vrha ili -1 ako je stog prazan i ima prototip

```
int pop ();
```

što će biti na stogu nakon obavljanja sljedeće naredbe, uz pretpostavku da je stog bio prazan i da stog raste s lijeva na desno:

```
push (push (pop ()) );
```

- a) 1 1
 - b) -1 1
 - c) 1
 - d) 0
 - e) stog će biti prazan
-

STOG-06

Funkcija za dodavanje elementa na stoga realiziran listom glasi:

a)

```
atom *dodaj (atom *vrh, int element) {
    atom *novi;
    if ((novi = (atom *) malloc(sizeof(atom))) != NULL) {
        novi->element = element;
        novi->sljed = vrh;
    }
    return novi;
}
```

b)

```
atom *dodaj (atom *vrh, int element) {
    atom *novi;
    novi->element = element;
    novi->sljed = vrh;
    return novi;
}
```

c)

```
int dodaj (atom *vrh, int element) {
    atom *novi;
    if ((novi = (atom *) malloc(sizeof(atom))) != NULL) {
        novi->element = element;
        novi->sljed = vrh;
        return 1
    }
    else
        return 0;
}
```

d)

```
atom *dodaj (atom *vrh, int element) {
    atom *novi;
    if ((novi = (atom *) malloc(sizeof(atom))) != NULL) {
        novi = element;
    }
    return novi;
}
```

e)

```
int dodaj (atom *vrh, int element) {
    atom *novi;
    if ((novi = (atom *) malloc(sizeof(atom))) != NULL) {
        novi = element;
        return 1
    }
    else
        return 0;
}
```


Odgovori

SORT-01: d)
SORT-02: d)
SORT-03: b)
SORT-04: c)
SORT-05: e)
SORT-06: a)

OO-01: e)
OO-02: e)
OO-03: b)
OO-04: b)
OO-05: a)
OO-06: a)

STOG-01: a)
STOG-02: b)
STOG-03: d)
STOG-04: b)
STOG-05: b)
STOG-06: a)