Strukture odvojite u zasebni header fajlove.

# Zadatak 1 - Datum

1. Deklarisati strukturu **Datum** sa atributima d, m i g.
2. Implementirati funkciju koja ispisuje podatke o datumu **date**

void info(Datum date)

1. Implementirati funkciju koja vraća objekat tipa Datum sa vrijednostima učitanih sa tastature

Datum ucitaj\_datum()

|  |
| --- |
| **Rješenje** |
|  |

# Zadatak 2 – Osoba

1. Deklarisati strukturu **Osoba** sa atributima id, ime.
2. Implementirati funkciju koja ispisuje podatke o osobi **o**

void info(Osoba o)

|  |
| --- |
| **Rješenje a i b** |
|  |

1. Alocirajte 5 objekatat (kao globalne varijable) tipa osoba (o1, o2, o3, o4, o5) i dodjelite neke vrijednosti.
2. Alocirajte niz tipa osoba veličine 5.

const int max\_s = 5;

Osoba studenti[max\_s] = {o1, o2, o3, o4, o5}

|  |
| --- |
| **Rješenje c i d – sa varijablama o1, ... o5** |
|  |
| **Rješenje c i d – bez varijabli o1, ... o5** |
|  |

1. Implementirati funkciju koja vraća objekat iz niza **studenti** na osnovu rednog broja kojeg korisnik unese pomoću tastature.

|  |
| --- |
| Spisak osoba:  1. Osoba A B, ID=302  2. Osoba C B, ID=303  3. Osoba D B, ID=304  4. Osoba E B, ID=305  5. Osoba F B, ID=306  Unesite RB. |

Osoba odaberite\_osobu()

|  |
| --- |
| **Rješenje e** |
|  |

1. U funkciji main implementirajte sljedeće

* Deklarište datume **d1**, **d2 i d3** i učitajte vrijednosti pomoću funkcije ucitaj\_datum
* Ispišite datume **d1, d2 i d3** pomoću funkcije info(datum)
* Odaberite studenta s1 i ispišite podatke pomoću funkcije info(osoba)

|  |
| --- |
| **Rješenje f** |
|  |

Prepravite prethodno rješenje

1. Prepravite funkciju odaberite\_osobu tako funkcija koristi različite nizove tipa osoba (npr. ispitivače i studente)

Osoba odaberite\_osobu() -> Osoba odaberite\_osobu(Osobe osobe[], int v)

|  |
| --- |
| **Rješenje g** |
|  |

# Zadatak 3 - Ispit

1. Implementirajte strukturu Ispit

|  |
| --- |
| struct Ispit  {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  };  Ispit ucitaj\_ispit();  void prikazi\_ispite(Ispit ispiti[], int max); |

|  |
| --- |
| **Rješenje** |
|  |

Program za testiranje

|  |
| --- |
| void main()  {  int \_brojac\_ispita = 0;  const int max\_ispita = 30;    Ispit ispiti[max\_ispita];  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  ispiti[\_brojac\_ispita++] = ucitaj\_ispit();  Ispit& i = odaberite\_ispit(ispiti, \_brojac\_ispita); //korisnik bira ispit  i.datum\_ispita.d++; //pomjeri datum ispita na naredni dan  prikazi\_ispite(ispiti, \_brojac\_ispita);  } |

# Zadatak 4 – Prijava za ispit

Dodajte strukturu **PrijavaZaIspit**.

U strukturu Ispit dodajte niz tipa **PrijavaZaIspit**

|  |
| --- |
| struct PrijavaZaIspit  {  Osoba student;  Datum datum\_prijave;  Datum datum\_odjave;  int ocjena;  };  const int max\_p = 100;  struct Ispit  {  Osoba ispitivac;  Datum datum\_ispita;  char predmet[10];  int brojac\_prijavljenih;  PrijavaZaIspit prijave[max\_p];  }; |

Prilagodite funkcijevoid ucitaj\_ispit() i void prikazi\_ispite(Ispit[], int) shodno novim modifikacijima u strukturi Ispit.

|  |
| --- |
| Rješenje |
|  |

# Zadatak 5 – Glavni program

Dovršite program

|  |
| --- |
| void main()  {  const int max\_ispiti = 10;  Ispit ispiti[max\_ispiti];  int brojac\_ispita = 0;  int x;  do  {  cout << "1. Ispiti - prikazi sve" << endl;  cout << "2. Ispiti - dodaj novi (zakazi novi ispit)" << endl;  cout << "3. Prijavljeni za ispit: prikazi sve (za ispit x)" << endl;  cout << "4. Prijavljeni za ispit: dodaj novu prijavu (za ispit x)" << endl;  cout << "5. Evidentiraj sve ocjene (za ispit x)" << endl;  cout << "0. EXIT" << endl;  cin >> x;  if (x == 1)  {  }  if (x == 2)  {    if (x == 3)  {  }  if (x == 4)  {  }  if (x == 5)  {  }  } while (x != 0);  } |

**Rješenje**

|  |
| --- |
|  |