

## Kompilacija programskih jezika - Septembar 2014.

1. Napisati program koji omogućava korisniku rad sa konačnim skupovima brojeva.

- (a) Skupovi se zadaju bilo navođenjem elemenata između { i }, bilo navođenjem intervala oblika a..b. Komandom `print` se ispisuju elementi skupa.

```
A = {3, 2, 2, -1};
B = 7..10;
print A;           {-1, 2, 3}
print B;           {7, 8, 9, 10}
```

- (b) Nad skupovima je moguće vršiti operacije unije (`\|`), preseka (`/\`) i razlike (`\`).

```
A = {1, 2, 3, 4};
B = {3, 5, 7};
print A \| B;      {1, 2, 3, 4, 5, 7}
print A /\ B;      {3}
print A \ B;       {1, 2, 4}
```

- (c) Program treba da omogući i proveru da li je dati broj element skupa (`:`) i da li je jedan skup podskup drugog (`<`).

```
check 5 : A;        False
check 7 : A \| B;    True
check A < B;        False
check {5, 7} < B;    True
```

## Kompilacija programskih jezika - Septembar 2014.

1. Napisati program koji omogućava korisniku rad sa konačnim skupovima brojeva.

- (a) Skupovi se zadaju bilo navođenjem elemenata između { i }, bilo navođenjem intervala oblika a..b. Komandom `print` se ispisuju elementi skupa.

```
A = {3, 2, 2, -1};
B = 7..10;
print A;           {-1, 2, 3}
print B;           {7, 8, 9, 10}
```

- (b) Nad skupovima je moguće vršiti operacije unije (`\|`), preseka (`/\`) i razlike (`\`).

```
A = {1, 2, 3, 4};
B = {3, 5, 7};
print A \| B;      {1, 2, 3, 4, 5, 7}
print A /\ B;      {3}
print A \ B;       {1, 2, 4}
```

- (c) Program treba da omogući i proveru da li je dati broj element skupa (`:`) i da li je jedan skup podskup drugog (`<`).

```
check 5 : A;        False
check 7 : A \| B;    True
check A < B;        False
check {5, 7} < B;    True
```

- (d) Kardinalnost skupova se izračunava korišćenjem komande `card`.

```
card {1, 3, 2, 7, 3, 4};          5
```

- (e) Definirati operaciju komplementiranja  $\sim$ . Komplementiranje se vrši u odnosu na univerzalni skup koji se zadaje kao `UniversalSet`.

```
UniversalSet = 1..10;
A = {1, 3, 5, 7, 9};
print ~A;          {2, 4, 6, 8, 10}
```

Ukoliko univerzalni skup nije prethodno definisan, program treba da prijavi grešku.

- (f) Komandom `partitive_set` se ispisuje partitivni skup datog skupa:

```
A = {1, 2};
partitive_set A;          {{}, {1}, {2}, {1, 2}}
```

- (g) Prilikom ispisivanja skupa, potrebno je da njegovi elementi budu iznova određeni na osnovu tekućih vrednosti promenljivih.

```
A = {1, 2};
B = {2, 3};
S = (A \ B) \ (B \ A);
print S;          {1, 3}
B = {1, 4};
print S;          {2, 4}
```

- (d) Kardinalnost skupova se izračunava korišćenjem komande `card`.

```
card {1, 3, 2, 7, 3, 4};          5
```

- (e) Definirati operaciju komplementiranja  $\sim$ . Komplementiranje se vrši u odnosu na univerzalni skup koji se zadaje kao `UniversalSet`.

```
UniversalSet = 1..10;
A = {1, 3, 5, 7, 9};
print ~A;          {2, 4, 6, 8, 10}
```

Ukoliko univerzalni skup nije prethodno definisan, program treba da prijavi grešku.

- (f) Komandom `partitive_set` se ispisuje partitivni skup datog skupa:

```
A = {1, 2};
partitive_set A;          {{}, {1}, {2}, {1, 2}}
```

- (g) Prilikom ispisivanja skupa, potrebno je da njegovi elementi budu iznova određeni na osnovu tekućih vrednosti promenljivih.

```
A = {1, 2};
B = {2, 3};
S = (A \ B) \ (B \ A);
print S;          {1, 3}
B = {1, 4};
print S;          {2, 4}
```