

- **Del 1** består av 8 flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt rätt antal alternativ får man 0 poäng på frågan.
  - **Del 2** består av ett antal frågor med varierande antal poäng vilka ska lösas genom att man skriver kod i de olika programmeringsspråken i kursen.
  - **Skriv tydligt.** Svårlästa svar riskerar 0 poäng.
  - Inga externa bibliotek får användas om det inte står explicit i uppgiften.
  - Skriv bara på en sida av varje papper.
  - För att få godkänt måste man ha minst 4 poäng på Del 1, har man inte det rättas inte Del 2.
  - **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
  - **Betygsgränser:** E: 15, D: 18, C: 21, B: 24, A: 27, av maximala 30.
- 

## Del 1: flervalsfrågor (1p per fråga, 8p totalt)

Var snäll och saml svaren på del 1 på ett svarspapper.

1. Vad blir resultatet av om man kör följande Prologkod: `1 + 1 is 2`.
  - A. true.
  - B. false.
  - C. `X = 1 + 1, Y = 2`.
  - D. `1 + 1 = 2`.
  - E. Det finns inget som heter "is" i Prolog
2. Vilka termer hänger specifikt ihop med funktionell programmering i Haskell?
  - A. Lathet
  - B. Pekararitmetik
  - C. Högre ordningens funktioner
  - D. Arv
  - E. Rekursion
3. Vad är typen på Haskellfunktionen:  

```
foo f g xs = map (f . g) xs
```

  - A. `(b -> c) -> (a -> b) -> [a] -> [c]`
  - B. `(a -> b) -> (b -> c) -> [a] -> [c]`
  - C. `(a -> b) -> [a] -> [b]`
  - D. `([a] -> b) -> ([b] -> c) -> [a] -> [c]`
  - E. Inget då koden leder till ett typfel.
4. Vilka av följande påståenden stämmer för JavaScript?
  - A. JavaScript är dynamiskt typat.
  - B. Variabler i JavaScript kan byta typ efter att man deklarerat dem.
  - C. JavaScript är statiskt typat.
  - D. JavaScript är en variant av Java.
  - E. Typfel hittas vid kompilering av JavaScript kod.

5. Vad kommer skrivas ut om man kör C-koden till höger?

A. 1

B. 2

C. 3

D. %d

E. Det går inte att säga då \*(p+2) pekar utanför arrayen.

```
int x[3] = {1,2,3};  
  
int *p = &x[0];  
  
printf("%d", *(p+2));
```

6. Vad gäller för Java koden till höger?

A. first\_name och last\_name är instansattribut

B. fn och ln är instansattribut

C. first\_name och last\_name är metoder

D. Givet en Person p kan man komma åt first\_name genom att skriva p.first\_name

E. full\_name är en metod

```
class Person {  
    private String first_name, last_name;  
  
    public Person(String fn, String ln) {  
        this.first_name = fn;  
        this.last_name = ln;  
    }  
  
    public String full_name() {  
        return first_name + " " + last_name;  
    }  
}
```

7. Hur många lösningar kommer Prolog generera för ett anrop till p(X,Y) givet koden nedan till höger?

A. 2

B. 5

C. 6

D. 7

E. 8

```
p(X,Y):- q(X), r(Y).  
p(0,0).  
  
q(1).  
q(2).  
q(3).  
  
r(4).  
r(5).
```

8. Vad blir resultatet av att köra Haskellkoden:

```
filter (\(x,y) -> x - y > 0) [(1,1),(1,2),(2,1)]
```

A. [(1,1)]

B. [(1,1),(1,2)]

C. [0,-1,1]

D. [(2,1)]

E. [False,False,True]

## Del 2: kodfrågor (22p totalt)

Var snäll använd ett papper till varje uppgift i del 2. Lösningarna på deluppgifterna för varje programmeringsspråk kan skrivas på samma papper.

### 9. Imperativ programmering i C

På tentan 2021-03-15 skulle man implementera en funktion för att kryptera strängar med hjälp av en matris med bestämt antal kolumner. Bokstäverna fylldes i rad för rad och sen läste man ut det krypterade ordet genom att läsa kolumnvis.

**Exempel:** för att kryptera Secret text med 3 kolumner skapar vi matrisen:

|   |   |   |
|---|---|---|
| S | e | c |
| r | e | t |
|   | t | e |
| x | t |   |

Den krypterade texten är då Sr xeettcte.

Nedan följer en exempellösning:

```

1 void matrix_encrypt(char* s, int n, int cols)
2 {
3     // calculate how many rows the matrix should have
4     // we have to take into account if cols divides n or not
5     int rows = n % cols == 0 ? n / cols : n / cols + 1;
6
7     for(int i = 0; i < cols; i++) {
8         for(int j = 0; j < rows; j++) {
9             // calculate the index to fetch
10            int k = j * cols + i;
11
12            // we should only fetch the index if it's not out of bounds
13            if (k < n)
14                printf("%c",s[k]);
15        }
16    }
17
18    printf("\n");
19 }

```

Uppgiften är nu att skriva om koden enligt instruktionerna nedan. Det räcker att skriva om koden med for loopar (dvs raderna 7–16).

- (a) Skriv om looparna så de är while loopar istället för for loopar. (2p)
- (b) Skriv om looparna med hjälp av goto istället för for loopar. (3p)

## 10. Objektorienterad programmering i Java

Den här uppgiften går ut på att skapa en klass MatrixEncrypter för matriskryptering på liknande sätt som ovan.

- (a) Klassen MatrixEncrypter ska ha privata instansattribut rows och cols av typ int. Klassen ska även ha en konstruktormetod som låter användaren sätta både rows och cols. (2p)
- (b) Skriv en publik metod public void encrypt(String s) som tar in en sträng s och skriver ut strängen krypterad med hjälp av en matris med storlek rows \* cols. Om strängen är så lång att den inte får plats i matrisen ska istället ett felmeddelande skrivas ut. (3p)

**Tips:** strängmetoderna charAt och length kan vara användbara.

**Exempelpörning:** om man testat klassen med koden

```

class uppgift10 {
    public static void main(String[] args) {
        // Create a matrix with 4 rows and 3 columns
        MatrixEncrypter m = new MatrixEncrypter(4,3);
        m.encrypt("Secret text");
        m.encrypt("Very long secret text");
    }
}

```

så ska utskriften bli:

```

Sr xeettcte
Error: the string is too long for the matrix

```

## 11. Funktionell programmering i Haskell

- (a) På ordinarie tenta skulle man skriva en rekursiv funktion `partition :: (a -> Bool) -> [a] -> ([a], [a])` som tar in en funktion `p` som returnerar `True` eller `False`, samt en lista `xs`. Funktionen skulle sedan returnera ett par bestående av en lista med de element i `xs` som gör `p` sann och en lista med de element som gör `p` falsk:

```
> partition (\x -> x > 3) [5,8,4,3,10,20]
([5,8,4,10,20],[3])
> partition (\x -> x > 3) [2,5,8,4,3,10,20]
([5,8,4,10,20],[2,3])
```

Skriv denna funktion med hjälp av den inbyggda Haskellfunktionen `filter`. (2p)

**Obs:** för poäng måste man använda sig av `filter` på ett lämpligt sätt.

- (b) Betrakta följande Haskelltyp för att representera filer på en dator:

```
data Files = File String | Folder String [Files]
```

En samling filer är alltså antingen ett filnamn eller en mapp innehållande fler filer/mappar. Uppgiften är nu att skriva funktionerna:

- `find :: String -> Files -> Bool` som ska söka efter ett filnamn i en samling filer. (2p)
- `folders :: Files -> [String]` som ska returnera alla mappnamn bland en samling filer. (2p)

**Exempelkörning:**

```
> fs = Folder "my_files" [ File "exam.pdf"
                          , Folder "songs" [ File "Born in the USA.mp3"
                                              , File "The river.mp3" ] ]
> find "The river.mp3" fs
True
> find "Dancing in the dark.mp3" fs
False
> folders fs
["my_files", "songs"]
```

## 12. Logikprogrammering i Prolog

Den här uppgiften går ut på att implementera insättningssortering i Prolog.

- (a) Skriv ett predikat `insert(X,XS,YS)` där `YS` är `XS` med `X` insatt på rätt position. (3p)

**Obs:** man kan anta att listan `XS` är sorterad så att det finns en korrekt plats för `X` att sättas in på.

**Exempelkörning:**

```
?- insert(2,[1,3],YS).
YS = [1, 2, 3] .
?- insert(4,[],YS).
YS = [4] .
```

- (b) Skriv `sort_list(XS,YS)` med hjälp av `insert` så att `YS` är den sorterade versionen av `XS`. (3p)

**Obs:** för poäng måste man använda `insert` och man får inte använda några inbyggda predikat.

**Exempelkörning:**

```
?- sort_list([4,3,2,5,1],YS).
YS = [1, 2, 3, 4, 5] .
```