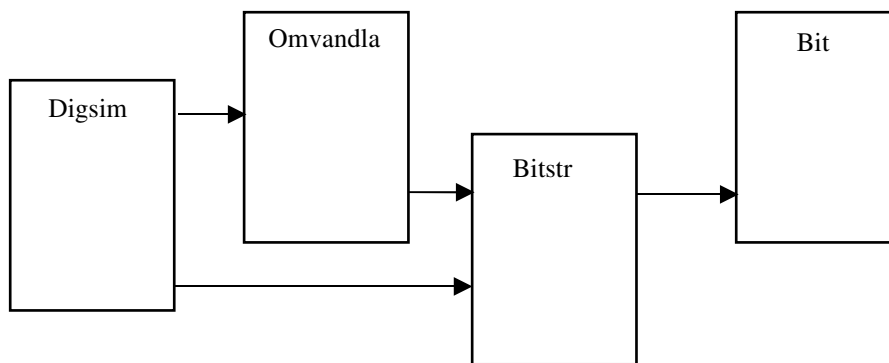


## Inlämningsuppgift 1, Digsim

### Specifikation

Skriv ett program som utför addition av heltal (positiva eller negativa) i binär form med hjälp av digitala grindar, som programmet ska simulera. Programmet ska läsa in två heltal, omvandla dessa till binär form (bitsträngar exempelvis "00010011"), addera bitsträngarna till en summasträng och slutligen omvandla summasträngen tillbaka till ett heltal. Antalet bitar i bitsträngarna ska läsas in varefter strängarna ska skapas dynamiskt. För att avbilda negativa heltal användes tvåkomplementmetoden. Det krävs ingen hantering av overflow, alltså då antalet bitar ej räcker till.

Programmet ska byggas i moduler enligt:

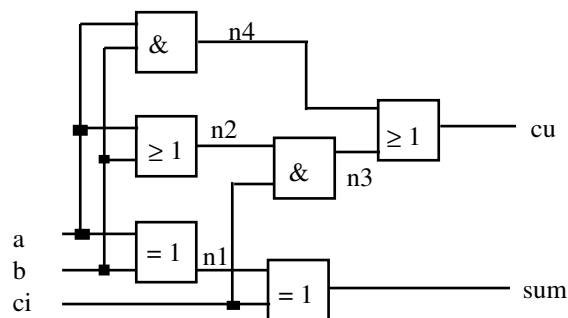


*Bit* Innehåller funktioner på bitnivå för de enkla digitala grindarna samt för en bitadderare enligt följande specifikation :

```
/* Bit.h */
```

```
char and(char a, char b); /* Returnerar '1' om a '1' och b '1' */  
char or(char a, char b);  
char xor(char a, char b);  
char not(char a);  
char nand(char a, char b);  
char nor(char a, char b);
```

```
void add(char a, char b, char ci, char *sum, char *cu);  
/* Adderar två bitar a och b med ci som in-carry och  
cu som ut-carry enligt figur */
```



*Bitstr* Innehåller funktioner för bitsträngar. Specifikationen ser ut som :

```
/* Bitstr.h */

void addstr(char *astr, char *bstr, char *sumstr);
/* Adderar astr och bstr till sumstr */

void twokomp(char *str);
/* Tvåkomplementerar str */
```

*Omvandla* Innehåller funktioner för omvandling mellan heltal och bitsträngar. Specifikation enligt :

```
/* Omvandla.h */

void tal_str(int tal, char *str, int nrbits);
/* Omvandlar tal till en binär sträng med nrbits bitar */

int str_tal(char *str);
/* Omvandlar den binära strängen str till ett heltal */
```

*Digsim* Huvudprogram som upprepat (avslut med 0) läser in antal bitar som du ska använda och dynamiskt allokera så långa strängar för den binära formen. Programmet ska sedan läsa in de båda heltalen, skriva ut deras binära representation, utföra additionen och skriva ut summan både i binär form och som heltal. Glöm ej att strängarna ska ha plats för '\0'-tecken och att avallokera minne.

*Extra* Komplettera modulen bitstr med en funktion för multiplikation av binära strängar och komplettera huvudprogrammet med multiplikation av heltal. (Ej obligatoriskt)

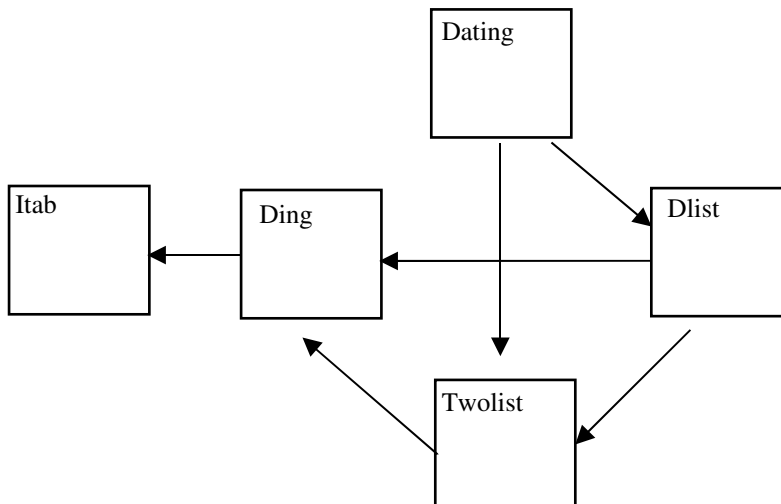
## Redovisning

Redovisning sker genom att programmet uppvisas och provkörs för läraren. Godkänd redovisning utgör en del av delkurs 2 i kursen Programmeringsmetodik.

## Inlämningsuppgift 2, Dating

Skriv ett program som kopplar ihop flickor med pojkar utgående ifrån deras gemensamma intressen. Programmet ska läsa ett antal flick- och pojknamn från filerna girlfile.txt resp boyfile.txt, som finns i din katalog och till varje flicka och pojke slumpa ett antal av intressena bio, dans, musik, idrott, resor och djur. Efter slumpandet ska flickorna instoppas i en flicklista och pojkarna i en pojklista. Programmet ska sedan fråga efter antalet gemensamma intressen och för varje flicka i flicklistan försöka hitta en pojke i pojklistan med minst detta antal lika intressen och sedan, om partner hittad, stoppa in paret sorterade efter flicknamn i en parlista och ta bort motsvarande flicka från flicklistan och pojke från pojklistan. Alla listor ska vara tvåvägslistor.

Programmet ska byggas av abstrakta datatyper enligt beroendediagrammet:



## *Itab*

```
/* Specifikation av intressetabeller -- itab.h */

#ifndef ITABH
#define ITABH

typedef enum {bio, dans, musik, idrott, resor, djur} intresstetyp;
typedef int itabtyp[6];

void slumpa(itabtyp itab);
/* Slumpar en intressetabell itab */

void visa_itab(itabtyp itab);
/* Visar intressetabellen itab på skärmen */

int antal_lika(itabtyp itab1, itabtyp itab2);
/* Returnerar antal lika intressen i itab1 och itab2 */

#endif
```

## *Ding*

```
/* Specifikation av dingar (flickor, pojkar, par) -- ding.h */

#ifndef DINGH
#define DINGH

#include "itab.h"

typedef struct
{
    char namn[30];
    itabtyp itab;
} persontyp;

typedef struct
{
    persontyp person;
    int is_pair;
    persontyp par_pojke;
} dingtyp;

void skapa_ding(dingtyp *ding, char *namn);
/* Skapar pojke eller flicka med namnet namn */

void skapa_par(dingtyp *par, dingtyp flicka, dingtyp pojke);
/* Skapar par av pojke och flicka */

void visa_ding(dingtyp ding);
/* Visar ding på skärmen */

int passar_ihop(dingtyp flicka, dingtyp pojke, int antal);
/* Kontrollerar om pojke och flicka har minst antal st lika intressen */

int is_less(dingtyp d1, dingtyp d2);
/* Jämför d1 och d2 med avseende på personnamnen */

#endif
```

## *Dlist*

```
/* Specifikation av datinglistor -- dlist.h */

#ifndef DLISTH
#define DLISTH

#include "twolist.h"
#include "ding.h"

void fyll_lista(headtyp *lista, char *filnamn);
/* Läser namn från fil, skapar ding och stoppar in ding i lista */

void para(headtyp *flicklist, headtyp *pojklst, headtyp *parlist, int
nr);
/* Letar för varje flicka i flicklist efter en pojke med minst nr
gemensamma intressen i pojklst och om det lyckas bildas ett par som
stoppas in i parlist och flickan och pojken tas bort från resp. lista */

void visa_lista(headtyp *lista);
/* Skriver ut listan på skärmen*/

#endif
```

## *Dating*

```
/* Huvudprogrammet - dating.c */
```

Ska börja med att skapa tomma tvåvägslistor för flickor, pojkar och par samt sedan försätta med en meny enligt:

fill lista     -- med undermeny för vilken lista flick eller pojk som ska fyllas

para           -- fyller parlistan med hittade par sorterade efter flicknamn

visa           -- med undermeny för vilken lista som ska visas på skärmen

rensa          -- med undermeny för vilken lista som ska rensas

sluta          -- avslutar programmet

Menyn ska ha filter för säker inmatning.

### *Extra*

Ej obligatorisk uppgift : Komplettera med ett alternativ ompara som frågar efter en flicka i parlistan och parar om henne med en annan pojke med ett nytt inläst antal lika intressen. Den först parade pojken stoppas tillbaka i pojklistan om ny pojke hittats.

### **Redovisning**

Uppgiften ska redovisas med en fullständig rapport innehållande försättsblad, innehållsförteckning, sammanfattning med användarnamn och katalog där den körbara filen finns samt bilagor i form av användaranvisning (manual), exempel på utskrift av en parlista som för 4 gemensamma intressen och den källkod som du skrivit.

Godkänd rapport utgör en del av delkurs 2 i kursen Programmeringsmetodik.