Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA

Tentamen
EXEMPEL flervalsfrågor

EXEMPEL

Utformningen av tentamen kommer att innehålla så kallade flervalsfrågor.

Denna tes ger exempel på hur utformningen av sådana flervalsfrågor kan se ut.

Observera att inget övrigt ändras, det är samma uppgiftstyper som tidigare och samma betygsgränser.

Du lämnar in svar på flervalsfrågor på en svarsblankett (sist i tesen) Fyll i denna och lämna in tillsammans med övriga lösningsblad. Du ska INTE lämna någon annan typ av lösning på dessa

Uppgift 1 (-2..6p)

Följande deklarationer är givna på "toppnivå".

```
static short a,b;
static int index;
static int vi[100];
```

Vi har nu tilldelningen:

```
vi[ index ] = (a+b) * (a-b);
```

Ange det alternativ (A-H) som visar en korrekt evaluering av hur uttrycken kodas i ARM v6 assemblerspråk. Om flera lösningar ger samma resultat ska du ange den kortaste.

A

```
LDR   R0,a
LDR   R1,b
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,vi
LDR   R0,index
LSL   R0,R0,#2
ADD   R0,R1,R2
STR   R2,[R0]
```

B

```
LDR   R0,a
LDR   R1,b
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,vi
LDR   R0,index
LDR   R0,[R0]
ADD   R0,R1,R2
STR   R2,[R0]
```

C

```
LDR   R0,a
LDR   R1,b
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,=vi
LDR   R0,=index
LDR   R0,[R0]
LSL   R0,R0,#2
ADD   R0,R1,R2
STR   R2,[R0]
```

D

```
LDR   R0,=a
LDRH  R0,[R0]
SXTH  R0,R0
LDR   R1,=b
LDRH  R1,[R1]
SXTH  R1,R1
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3,R3
LDR   R1,=vi
LDR   R0,=index
LDR   R0,[R0]
LSL   R0,R0,#2
ADD   R0,R1,R2
STR   R2,[R0]
```

E

```
LDR   R0,=a
LDRH  R0,[R0]
SXTB  R0,R0
LDR   R1,=b
LDRH  R1,[R1]
SXTB  R1,R1
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,=vi
LDR   R0,=index
LDR   R0,[R0]
LSL   R0,R0,#2
ADD   R0,R1,R2
STR   R2,[R0]
```

F

```
LDR   R0,=a
LDRH  R0,[R0]
SXTH  R0,R0
LDR   R1,=b
LDRH  R1,[R1]
SXTH  R1,R1
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,=vi
LDR   R0,=index
LDR   R0,[R0]
LSL   R0,R0,#2
ADD   R0,R1,R0
STR   R2,[R0]
```

G

```
LDR   R0,a
LDR   R1,b
ADD   R2,R0,R1
SUB   R3,R0,R1
MUL   R2,R3
LDR   R1,=vi
LDR   R0,=index
LDR   R0,[R0]
LSL   R0,R0,#2
ADD   R0,R1,R2
STR   R2,[R0]
```

H

Inget av lösningsförslagen är korrekt

Uppgift 2 (-2..8p)

För ett fält med tecken (`char`) kan man definiera en operation *rotation vänster* på fältet som en operation som placerar fältets första tecken på den sista platsen i fältet, det andra tecknet på den första platsen, det tredje tecknet på den andra platsen o.s.v.

En funktion `rotateleft` som roterar ett fält med tecken ett godtyckligt antal steg åt vänster har konstruerats av ett programmerarlag. Funktionen har tre parametrar, en pekare till fältet som skall roteras samt två heltal (`int`) vilka anger fältets längd respektive antalet rotationssteg. Parametrarna ges i denna ordning, funktionen har inget returvärde och ger heller inga sidoeffekter.

Ange det alternativ (A-I) som visar en korrekt implementering (hur funktionen kodas) i C.

A

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (;steps > 0; steps--) {
    next++;
    current  = f;
    i = l;
    save = *current;
    while( i>1 )
    {
      *current-- = *next--;
      i--;
    }
    *current = save;
  }
}
```

B

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (;steps > 0; steps--) {
    next++ = current  = f;
    i = l;
    save = *current;
    while( i )
    {
      *current++ = *next++;
      i--;
    }
    *current = save;
  }
}
```

C

```
void rotateleft(char* f,
    int l, int steps) {
    int i;
    char* current, * next, save;
    for (; steps > 0; steps--) {
        next = current = f;
        next += 1;
        i = l;
        save = *current;
        while (i > 1)
        {
            *current = *next++;
            current++;
            i--;
        }
        *current = save;
    }
}
```

D

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (;steps > 0; steps--) {
    next = current  = f;
    i = l;
    save = *current;
    while( i>1 )
    {
      *current++ = *next++;
      i--;
    }
    *current = save;
  }
}
```

E

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (;steps > 0; steps) {
    next++ = current  = f;
    i = l;
    save = *current;
    while( i )
    {
      *current++ = *next++;
      i--;
    }
    *current = save;
  }
}
```

F

Inget av lösningsförslagen är korrekt

G

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (;steps > 0; steps--) {
    next++ = current  = f;
    i = l;
    save = *current;
    while( i>1 )
    {
      *next-- = *current--;
      i--;
    }
    *current = save;
  }
}
```

H

```
void rotateleft(char *f,
       int l, int steps) {
  int i;
  char *current, *next, save;
  for (steps > 0; steps--) {
    next++ = current  = f;
    i = l;
    save = *current;
    while( i>1 )
    {
      *current++ = *next++;
      i--;
    }
    *current = save;
  }
}
```

I

Flera lösningsalternativ är korrekta

Svarsblankett

Anonym kod

Maskinorienterad programmering, tentamen flervalsfrågor.

Kryssa i det alternativ du anser vara korrekt, eller avstå.
Lämna in denna blankett tillsammans med dina övriga lösningar.

Fråga	A	B	C	D	E	F	G	H	I
1									
2									

Lösningsförslag

Uppgift 1 (6p)
Alternativ F.

Fullständigt lösningsförslag med kommentarer

```
    LDR   R0,=a      @ R0← &a
    LDRH  R0,[R0]    @ R0← a
    SXTH  R0,R0      @ R0← (int) a
    LDR   R1,=b      @ R1← &b
    LDRH  R1,[R1]    @ R1← b
    SXTH  R1,R1      @ R1← (int) b
    ADD   R2,R0,R1   @ R2← (int)a+(int)b
    SUB   R3,R0,R1   @ R3← (int)a-(int)b
    MUL   R2,R3      @ R2← (a+b)*(a-b)

    LDR   R1,=vi     @ R1← &vi
    LDR   R0,=index  @ R0← &index
    LDR   R0,[R0]    @ R0← index
    LSL   R0,R0,#2   @ R0← index*sizeof(int)
    ADD   R0,R1,R0   @ R0← &vi+ index*sizeof(int)
    STR   R2,[R0]    @ vi[index] ← ((int)a+(int)b) * ((int)a-(int)b)
```

Uppgift 3 (8p)
Alternativ C.

Fullständigt lösningsförslag med kommentarer

```
void rotateleft(char* f, int l, int steps) {
    int i;
    char* current, * next, save;
    for (; steps > 0; steps--) {
        next = current = f;
        next += 1;
        i = l;
        save = *current; // Kommer att bli det sista elementet
        while (i > 1)    // För alla element utom det sista: ...
        {
            *current = *next++; // Skifta elementet ett steg åt vänster
            current++;
            i--;
        }
        *current = save; // Lägg sista elementet på plats
    }
}
```