UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

MEng Honours Degrees in Computing Part IV
MEng Honours Degree in Information Systems Engineering Part IV
MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Diploma of Membership of Imperial College
Associateship of the City and Guilds of London Institute

PAPER 4.19 / I4.6

ADVANCED COMPILERS

Wednesday, April 23rd 1997, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 Consider the following fragments of a C++program:

```
class A {
public:
      int x;
      virtual void f();
      void h();          };

class B : public A {
public:
      int y;
      virtual void f();
      virtual void g(); };

class C : public B {
public:
    int z;
    virtual void f();
    void h();            };

// bodies for functions    A::f(), A::h(), B::f(),
//                         B::g(), C::f(), C::h()

A anA, *anAP1, *anAP2;
B aB, *aBP;
C aC, *aCP;
anAP1 = new A;
aBP   = new B;
aCP   = new C;
anAP2 = aCP;
```

a Sketch the representation of `anA`, `aB`, `AC`, `anAP1`, `anP2`, `ABP` and `ACP`.

b How are the following data members addressed (assuming that member functions have an implicit `thisX` parameter representing the address of the object executing the function, and that integers and addresses are stored in 4 bytes):

x	in the body of	`A::f()`
x	in the body of	`B::f()`
x	in the body of	`C::f()`
z	in the body of	`C::h()`

c Consider the following function calls:

```
anA.f();
anAP1->f();
anAP2->f();
aB.h();
anAP2->h();
```

i) Which functions are called? (Use the notation `A::f`, `B::f` *etc* .)

ii) Outline the corresponding code. (Use `&w` to represent the address of w.)

Question continued on next page ...

d Consider the program fragments:

```
class D {
public:
     int i;
     virtual void f();
     virtual int f(D);};

class E : public D {
public:
     virtual void f();
     virtual void f(A);};

 // bodies for functions    D::f(), D::f(D)
 //                         E::f(), E::f(A)

D aD;
E anE;
```

and sketch the representation of `aD` and `anE`. Remember that the functions `f()`, `f(D)`, `f(A)` overload each other.

Turn over ...

2a i What is an *induction variable* for a loop?

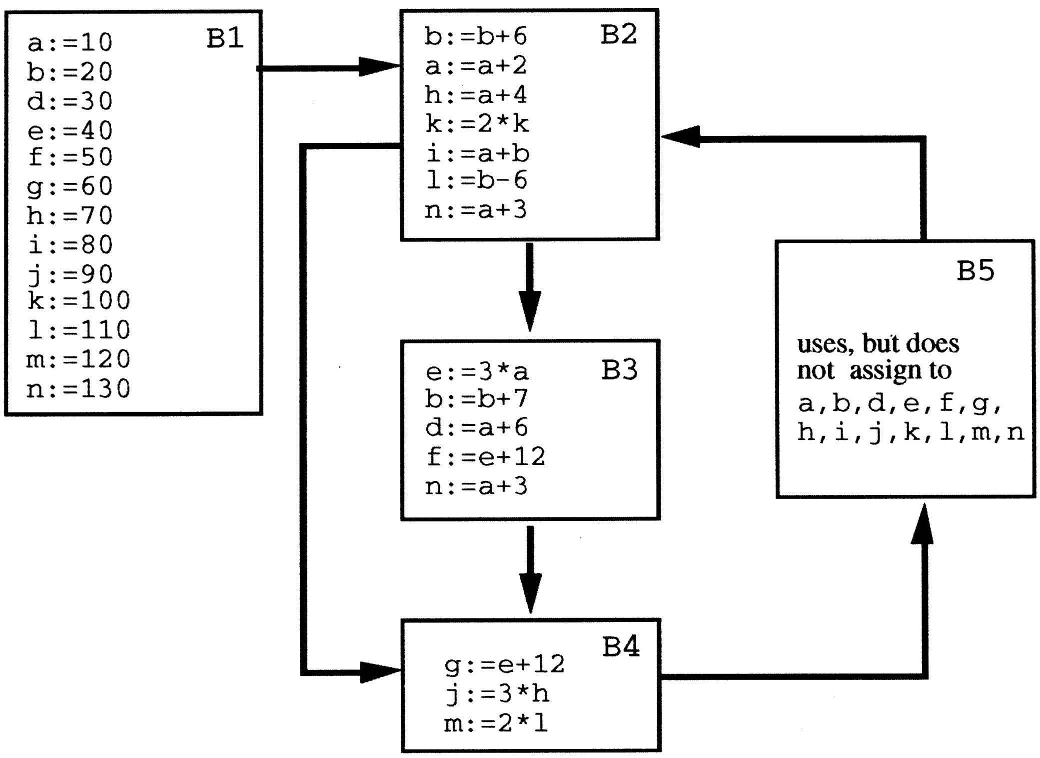
ii What is a *basic induction variable*?

iii What is a *non-basic induction variable*?

iv What information does the *triplet* of an induction variable `i` represent?

v Is an induction variable of an enclosed loop also an induction variable of an enclosing loop?

b The following flow diagram contains a loop consisting of blocks `B2`, `B3`, `B4` and `B5`. Block `B5` uses the variables `a`, `b`, `d`, `e`, `f`, `g`, `h`, `i`, `j`, `k`, `l` and `m`, but does not assign to any of these variables.



For each of the above variables, if it is an induction variable, give its triplet, otherwise briefly justify why it is not.

c i What is the objective of *strength reduction* ?

ii Draw the flow diagram corresponding to the application of strength reduction to the program shown in part b.

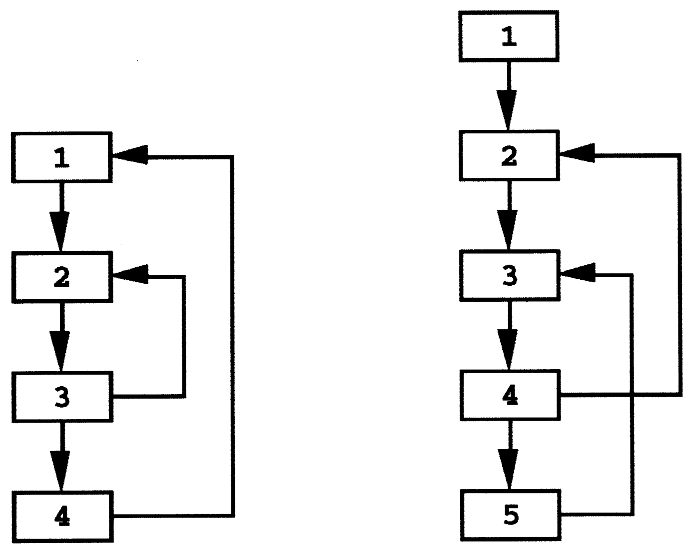
3a Give the definition of a *back edge* in a flow graph in terms of the *dominates* relationship. (Remember that a node `n1` dominates another node `n2` if every path from the initial node to `n2` goes through `n1`.)

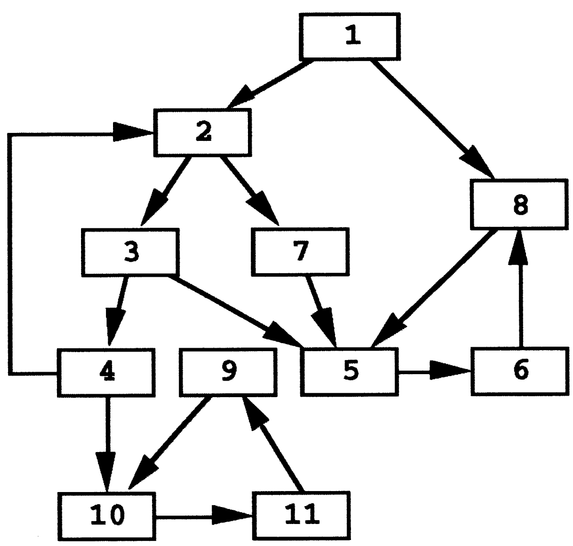
b Give the definition of a *natural loop* in terms of the back edge relationship.

c The *header* of a loop dominates all nodes in the loop. Justify.

d Any loop is *strongly connected*, *i.e.* for any two nodes `n1` and `n2` there exists a path from `n1` to `n2`. Justify.

e Consider the following three flow graphs:





For each of the graphs:

i Which are the back edges?

ii List the nodes belonging to the corresponding loops.

Turn over ...

4a Consider the following loop:

```
for i1 = 1 to N
  for i2 = 1 to M
    A[i1,i2] := A[i1-1,i2] + A[i1+1,i2]
```

i Sketch the iteration space for this loop, and mark all the dependences present.

ii There are two dependences to be considered here; write down the two dependence equations.

iii Write down the dependence distance vectors corresponding to each dependence. What kind of dependences are present?

iv Write down the unimodular transformation matrix representing reversal of the outer (`i1`) loop.

v Show using matrix-vector multiplication whether this transformation is valid, and explain your answer.

b Consider the following pair of loops (the first is the same as in part (a)):

```
for i1 = 1 to N
  for i2 = 1 to M
    A[i1,i2] := A[i1-1,i2] + A[i1+1,i2]
for i1 = 1 to N
  for i2 = 1 to M
    A[i1,i2] := A[i1,i2-1] + A[i1,i2+1]
```

i Write down the dependence distance vectors of the two dependences present in the second loop.

ii Is loop fusion possible here? Justify your answer carefully.

(The seven parts carry, respectively, 10%, 10%, 15%, 10%, 20%, 10% and 25% of the marks).

End of Paper