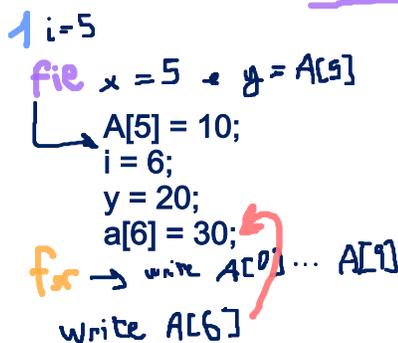


Tempo a disposizione: ore 2.

1. Si dica cosa stampa il seguente frammento di programma assumendo scope statico e passaggio dei parametri per nome.

```
int i = 1;
int[] A = new int[10];
void fie (int x name, int y name) {
    A[i] = 10;
    i = x++;
    y = 20;
    A[i] = 30 ;
}
for (j = 0; j <= 9; j+= 1)
    {A[j] = 0;}

{
    int i = 5;
    fie (i,A[i]);
    for (j = 0; j <= 9; j+= 1)
        {write(A[j] )}
    write (A[i] )
}
```

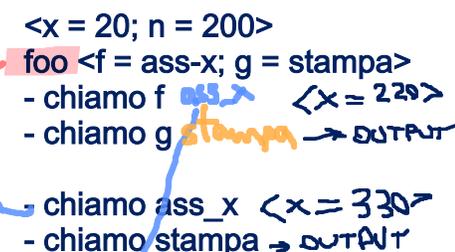


Stampa: $A[0] \dots A[9] A[6]$
 dove $A[5] = 10; A[6] = 30$

2. Si dica cosa stampa il seguente frammento di programma assumendo scope statico e deep binding:

```
{int x = 10;
 int n = 100;
 int ass_x(){
    x = x+n;
 }
 int stampa(){
    write(x);
 }
 void foo (int f(), int g()){
    int x = 30;
    int n = 300;
    f();
    g();
    ass_x();
    stampa();
 }
}

{int x = 20;
 int n = 200;
 int ass_x(){
    x = x+n;
 }
 foo(ass_x(), stampa());
}
```



Output:
 220
 330

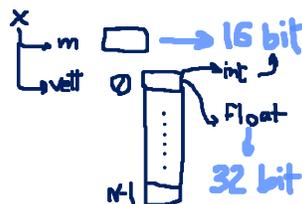
3. Si consideri il seguente codice dove il costrutto union permette di realizzare un record variante (analogamente al C)

```
const int N = 10;

union tipoCella{
    int cellaInt;
    float cellaFloat;
};

struct tipoVettore{
    int m;
    tipoCella vett[N];
};

struct tipoVettore x;
```



dunque

$$(16+32) \cdot (N-1) + (16+32)$$

$$= (16+32) \cdot N =$$

$$= 960 \text{ bit occupati}$$

Si descriva il layout di memoria di x assumendo una parola di 32 bit, allineamento alla parola, che un int occupi 16 bit un float occupi 32 bit.

4. Si descriva sinteticamente in cosa consiste, e quale problema risolve, la tecnica del reference count.

5. Si consideri l'implementazione dello scope statico mediante catena statica. Si dica, motivando le risposte: i) se la lunghezza massima della catena statica può essere determinata durante la compilazione; ii) se al posto dei puntatori di catena statica si potrebbe memorizzare un indirizzo costituito da un offset rispetto all'inizio delle pila nella quale sono memorizzati i record di attivazione.
6. Si dica motivando brevemente la risposta cosa viene stampato dal seguente frammento in un linguaggio con eccezioni

```

class X extends Throwable{
}
class Y extends X{
}

void f() throws X {
    throw new X();
}

void g (int sw) throws X , Y {
    if (sw == 0) {f();}
    try {f();} catch (X e) {write("in_g");}
}

...
main {
    try {g(1);}
        catch (Y e) {write("in_main");}
}

```

in g() il gestore delle eccezioni prova f() che lancia una eccezione Y, facendo stampare "in main".
 Ciò succede poiché l'eccezione catturata è una versione estesa di X e quindi non è gestibile dall'handler di g().
 Nel main riesce ad eseguire l'handler ed effettua la stampa.

7. E' dato il seguente programma Prolog (X,Y,Z,W sono variabili mentre a,b,c sono costanti).

```

p(a,b):- p(a,b).
p(b,b):- r(W).
p(c,c):- r(b)
r(a).

```

Si dica, giustificando brevemente le risposte, che risultati danno le valutazioni dei seguenti goal:

```

?p(Z,Z).
?p(Z,W).
?p(c,W).

```

8. Discutere brevemente se e come è possibile simulare la comunicazione sincrona mediante quella asincrona e vice versa.