

# Semplici traduzioni Java-C

E Mumolo

## myswap.java

```
import java.io.*;
class myvar{
    private int x;
    myvar(int a) {x = a;}
    int get() { return x; }
    void set(int a) {x = a;}
}
class myswap{
    static void swap(myvar a, myvar b)
    {
        int t = a.get();
        a.set(b.get());
        b.set(t);
    }

    public static void main(String argv[])
    {

        myvar aa=new myvar(47);
        myvar bb=new myvar(13);

        System.out.println("Prima:a="+aa.get()+" b="+bb.get());
        swap(aa,bb);
        System.out.println("Dopo:a="+aa.get()+" b="+bb.get());
    }
}
```

## myswap.c

```
#include <stdio.h>
void swap(int *i, int *j) {
    int t = *i;
    *i = *j;
    *j = t;
}
void main() {
    int a = 23, b = 47;
    swap(&a, &b);
    printf("a: %d, b: %d\n", a, b);
}
```

## uno.java

```
import java.io.*;
class P{
    private float constant;
    private String ptr;
    P(){};

    float getC() {return constant;}
    void setC(float newConstant) {
        this.constant=newConstant;
    }
    String getP() {return ptr;}
    void setP(String newPtr) {
        this.ptr=newPtr;
    }
}
class uno {
    public static void main(String[] args) {
        P variable = new P();
        String str = "Programmazione";

        variable.setC(1.7f);
        variable.setP(str);

        System.out.println(variable.getC());
        System.out.println(variable.getP());
    }
}
```

## uno.c

```
#include <stdio.h>

struct programming
{
    float constant;
    char *pointer;
};

int main()
{
    struct programming variable;
    char string[] = "Programmazione.";

    variable.constant = 1.23;
    variable.pointer = string;

    printf("%f\n", variable.constant);
    printf("%s\n", variable.pointer);

    return 0;
}
```

## due.java

```
import java.io.*;
class Name {
    private int a ;
    private float b ;
    private String c ;
    int getA() { return a;    }
    void setA(int newA) { a=newA; }
    float getB() { return b; }
    void setB(float newB) {b=newB;    }
    String getC() { return c;    }
    void setC(String newC) {c=newC;    }
}
class due {
    public static void main(String[] args) {
        Name[] ptr; int n;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Numero di strutture: ");
        n = scanner.nextInt();
        ptr = new Name[n];
        for(int i = 0; i < n; ++i) {
            ptr[i]=new Name();
            System.out.print("Scrivi stringa:");
            ptr[i].setC(scanner.next());
            System.out.print("Scrivi intero:");
            ptr[i].setA(scanner.nextInt());
            System.out.print("Scrivi float:");
            ptr[i].setB(scanner.nextFloat());
        }
        System.out.println("stampa tutto:");
        for(int i = 0; i < n; ++i) {
            System.out.printf("%s\t%d\t%.2f\n", ptr[i].getC(), ptr[i].getA(), ptr[i].getB());
        }
    }
}
```

# due.c

```
#include <stdio.h>
#include<stdlib.h>
struct name {
    int a;
    float b;
    char c[30];
};

int main(){
    struct name *ptr;
    int i,n;
    printf("Enter n: ");
    scanf("%d",&n);
    Ptr=(struct name*)malloc(n*sizeof(struct name));

    for(i=0;i<n;++i){
        printf("Dare stringa, intero e float:\n");
        scanf("%s%d%f",&(ptr+i)->c,&(ptr+i)->a,&(ptr+i)->b);
    }
    printf("Stampa tutto:\n");
    for(i=0;i<n;++i)
        printf("%s\t%d\t%.2f\n", (ptr+i)->c, (ptr+i)->a, (ptr+i)->b);
    return 0;
}
```

## tre.java

```

import java.io.*;
import java.util.Scanner;

class Name {
    private int a;
    private float b ;

    int getA() {return a;}
    void setA(int newA) {a=newA;}

    float getB() {return b;}
    void setB(float newB) {b=newB;}
}

public class tre{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Name[] ptr;
        Name[] p = {new Name()};
        ptr = p;

        System.out.print("scrivi intero: ");
        ptr[0].setA(scanner.nextInt());
        System.out.print("scrivi float: ");
        ptr[0].setB(scanner.nextFloat());
        System.out.print("stampa: ");
        System.out.printf("%d  %f\n", ptr[0].getA(), ptr[0].getB());
    }
}

```

## tre.c

```

#include <stdio.h>
struct name{
    int a;
    float b;
};
int main(){
    struct name *ptr,p;
    ptr=&p;
    printf("Enter integer: ");
    scanf("%d",&(*ptr).a);
    printf("Enter number: ");
    scanf("%f",&(*ptr).b);
    printf("Displaying: ");
    printf("%d  %f",(*ptr).a,(*ptr).b);
    return 0;
}

```

## quattro\_1.java

```
import java.io.*;
class quattro_1 {
    public static void main(String[] args) {
        int posB = 1, posC = 2, posD = 3;
        for(int pos = 0; pos<4; pos++)
            switch(pos) {
                default:
                    System.out.println("a");
                    continue;
                case 1:
                    System.out.println("b");
                    continue;
                case 2:
                    System.out.println("c");
                    continue;
                case 3:
                    System.out.println("d");
                    return;
            }
    }
}
```

## quattro\_1.c

```
#include <stdio.h>

int main()
{
    printf("a\n");
    goto c;
b:    printf("b\n");
    goto d;
c:    printf("c\n");
    goto b;
d:    printf("d\n");
    return 0;
}
```

```
public class str {                                     quattro_2.java
    public static int max = 4;

    public static void main(String[] args) {
        String names[];
        names = new String[] {"Giulio","Francesco","Valeria","Fabrizio"};

        int i = 0;

        for(i = 0; i < max; i++) {
            System.out.println("array[" + i + "] = " + names[i]);
        }
    }
}
```

```
-----
#include <stdio.h>                                     quattro_2.c
const int MAX = 4;

int main ()
{
    char *names[] = {    "Giulio",  "Francesco",  "Valeria",    "Fabrizio"  };
    int i = 0;
    for ( i = 0; i < MAX; i++)
    {
        printf("Value of names[%d] = %s\n", i, names[i] );
    }
    return 0;
}
```



```

import java.io.*;
class punto {
    private float x, y;
    punto(float a, float b){x=a; y=b;};
    punto(float a){ x=a; y=0; };
    punto(){ x=0; y=0; };
    float getx(){return x;}
    float gety(){return y;}
    void setx(float newx){x=newx;}
    void sety(float newy){y=newy;}
    float distanza(punto a){
        float t=(float)(Math.pow(x-a.x,2)+Math.pow(y-a.y,2));
        return (float)Math.sqrt(t);
    }
};
class cinque{
    static void out(float a, float b, float c, float d)
    {
        punto p=new punto(0,0); punto p2=new punto(3);
        punto q=new punto(a,b); punto p3=new punto();
        System.out.println("coordinate punto q="+q.getx() + " " + q.gety());
        p.setx(c); p.sety(d);
        System.out.println("coordinate punto p="+p.getx() + " " +p.gety());
        System.out.println("coordinate punto p2="+p2.getx() + " " +p2.gety());
        System.out.println("coordinate punto p3="+p3.getx() + " " +p3.gety());
        System.out.println("distanza tra p e q="+ q.distanza(p));
        System.out.println("distanza tra p2 e p3="+ p3.distanza(p2));
    }
    public static void main(String argv[])
    {
        out(1,2,5,5);
        System.exit(0);
    }
}

```

cinque.java

```

#include <stdio.h>
#include <math.h>
struct punto{
    float x;
    float y;
};
float distanza(struct punto a,struct punto b){
    return (float)sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));
}
int main(){
    struct punto p,q,p2,p3;
    float a,b;

    a=1.0; b=2.0;

    p.x=0.; p.y=0.; q.x=a ; q.y=b;
    p2.x=3; p2.y=0; p3.x=0.; p3.y=0.;

    printf("coordinate punto q=(%3.2f,%3.2f)\n", q.x, q.y);
    p.x=5.; p.y=5.;
    printf("coordinate punto p=%3.2f,%3.2f)\n", p.x, p.y);

    printf("coordinate punto p2=(%3.2f,%3.2f)\n", p2.x, p2.y);
    printf("coordinate punto p3=(%3.2f,%3.2f)\n", p3.x, p3.y);

    printf("distanza tra p e q=%3.2f\n", distanza(p,q));
    printf("distanza tra p2 e p3=%3.2f\n", distanza(p2,p3));
}

```

cinque.c

```
CC=gcc
CFLAGS=
RM=rm -rf
OUT=vector
```

File Makefile

```
all: build
```

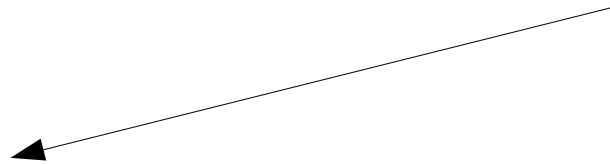
```
build: main.o vector.o
    $(CC) $(CFLAGS) -o $(OUT) main.o vector.o
    $(RM) *.o
```

```
debug: CFLAGS+=-DDEBUG_ON
debug: build
```

```
main.o: main.c vector.h
    $(CC) $(CFLAGS) -c main.c
```

```
vector.o: vector.c vector.h
    $(CC) $(CFLAGS) -c vector.c
```

```
clean:
    $(RM) *.o $(OUT)
```



# Programmi da fare in C utili per esercitarsi col linguaggio

- Considerando che:

```
int n;  
char buf[256];  
(n = read(STDIN_FILENO, buf, 256))
```

- è un frammento di programma C che legge una stringa di lunghezza massima 256 caratteri dallo standard input (tastiera) che viene messa in buf[]
- il valore restituito da read, 'n', e' il numero di caratteri effettivamente letto da tastiera
- se 'n' e' <0 vuol dire che c'e' stato un errore
- il frammento di programma che scrive sullo standard output (video) n caratteri e' il seguente:  

```
m=(write(STDOUT_FILENO, buf, n))
```
- dove m e' il numero di caratteri effettivamente scritti (se non ci sono problemi, m=n)
- Scrivere un programma C che scrive continuamente sullo standard output quello che l'utente scrive sullo standard input
- Chiamare questo programma mycat.c

## Programmi in C utili per esercitarsi col linguaggio

- Fare una modifica del programma precedente dove la stringa scritta sullo standard output viene modificata da una funzione di C che trasforma tutti i caratteri in maiuscolo
- Scrivere la funzione in un file separato e compilarlo separatamente
- La funzione usi puntatori per passare la stringa
  
- Scrivere un programma C che realizza questo pseudocodice:

```
while(true){
```

  - Scrivi \$ sul terminale
  - leggi la stringa scritta dall'utente
  - chiama una funzione che scrive le stringhe dell'utente in un array di stringhe, aggiornando l'indice (la 1a stringa viene scritta nella riga [0][], la seconda in [1][] e così via
  - Scrive sul terminale l'indice corrente

```
}
```
- Chiamare questo programma myshell.c