

```
/*questo programma descrive il comportamento del percettrone a due ingressi
e una uscita e calcola la funzione OR come uscita, la funzione di attivazione
e la sommatoria del prodotto dei pesi e degli ingressi e la funzione di trasferimento e la
funzione a gradino, quindi il percettrone ha solo due uscite o zero o uno,
in ingresso abbiamo 4 coppie che combinate ci danno i valori OR*/
```

```
#include<stdio.h>
//#include<stdlib.h>
//#include<cstdio.h>
//#include<cstdlib.h>
//#include<iostream.h>
//#include"pr9.h"

typedef double T_Precision;
```

```
T_Precision transfer_function(const T_Precision *x,const T_Precision *w, size_t n);
```

```
int main (void)
{
    const size_t n = 2; /*numero di ingressi*/
    T_Precision x[n+1]; /*vettore ingressi del percettrone piu il bias*/
    T_Precision w[n+1]; /*vettore pesi sinaptici piu il peso del bias*/
    T_Precision y; /*uscita del percettrone*/
    size_t i,j; /*iteratori*/
    x[n] = -1; /*valore di bias*/
    w[0] = 5; /*imposto valore dei pesi*/
    w[1] = 5;
    w[2] = 4; /*peso sinaptico del bias*/

    printf ("[x1]\t[x2]\t[y]\n"); /*stampo intestazione tabella*/

    for (j=0; j<=1; j++)
    { /*provo tutte le combinazioni degli ingressi,ossia permuti gli ingressi, se
gli ingressi sono solo coppie di zeri e uno ho un quadrato nello spazio degli
ingressi quindi 4 vertici */

        for (i=0; i<=1; i++)
        {

            x[0]=i;
            x[1]=j;

            y=transfer_function(x,w,n); /*calcolo l uscita del percettrone*/

            printf("%.0f\t%.0f\t%.0f\n",x[0],x[1],y); /*stampo gli ingressi e la
rispettiva uscita*/

        }

    }

    return 0;
} /*chiusura main*/
```

```
T_Precision activation_function(const T_Precision *x,const T_Precision *w, size_t n)
{
T_Precision potential = 0;
size_t i;

for (i=0;i<=n;i++)
{
potential += x[i]*w[i];
}
return potential;
}
```

```
T_Precision transfer_function(const T_Precision *x,const T_Precision *w, size_t n)
{
T_Precision activation_function(const T_Precision *x,const
T_Precision *w, size_t n);//dichiarazione

T_Precision potential = activation_function(x,w,n);
return (T_Precision) (potential>0);
}
```