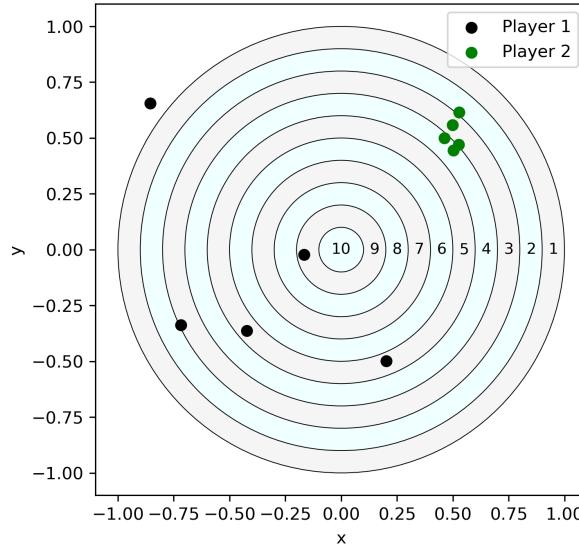


Un bersaglio per frecce è costituito da dieci cerchi concentrici. Di seguito, si considera un sistema di riferimento cartesiano con origine nel centro del bersaglio e viene omessa l'unità di misura delle lunghezze. Si supponga che il bersaglio abbia raggio 1 e che i cerchi concentrici abbiano raggio $0.1, 0.2, 0.3, \dots, 1$. Ogni corona circolare risultante è associata ad un punteggio che decresce all'aumentare della distanza dal centro del bersaglio. Alla corona compresa tra il centro e il raggio 0.1 (escluso) è associato il punteggio 10, mentre alla corona più esterna, compresa tra raggio 0.9 e raggio 1.0 (escluso), è associato il punteggio 1. Se una freccetta non colpisce il bersaglio, il tiro non è valido ed il punteggio associato è 0.

Nella figura seguente viene riportato un bersaglio nel sistema di riferimento cartesiano, assieme all'indicazione dei punteggi di ciascuna corona circolare. Inoltre è rappresentato l'esito di una partita di frecce tra due giocatori.



Nel corso della partita, i due giocatori si alternano nel tirare 5 frecce a testa. L'esito dei tiri dei due giocatori può essere rappresentato in Python come una lista di 10 tuple, ciascuna contenente 2 valori float che rappresentano le coordinate (x, y) di ciascun tiro nel sistema di riferimento cartesiano.

Con riferimento all'esempio in figura, la lista contiene i seguenti valori (si riportano solo le prime tre tuple per brevità).

```
partita = [(-0.16670, -0.02251), (0.52757, 0.61461), (-0.85448, 0.65611), ... ]
```

- Il primo tiro di *Player 1* ha coordinate $x = -0.16670, y = -0.02251$
- Il primo tiro di *Player 2* ha coordinate $x = 0.52757, y = 0.61461$
- Il secondo tiro di *Player 1* ha coordinate $x = -0.85448, y = 0.65611$

I punteggi ottenuti dai tiri di *Player 1* sono 9, 0, 3, 5, 5. I punteggi ottenuti dai tiri di *Player 2* sono 2, 4, 3, 3, 4.

1. Definire la funzione `calcola_distanza`

- **Parametri di ingresso:**
 - `x`: float.
 - `y`: float.
 - `x_ref`: float (default = 0).
 - `y_ref`: float (default = 0).
- **Restituisce:** un float
- **Descrizione:** la funzione restituisce la distanza euclidea tra il punto (`x,y`) e il punto (`x_ref, y_ref`).
- **Esempio:** Con riferimento all'esempio riportato nella pagina introduttiva, la distanza del primo tiro di *Player 1* dal centro del bersaglio è 0.16821.

2. Definire la funzione `calcola_punteggio_tiro`

- **Parametri di ingresso:**
 - `tiro`: tupla contenente due float.
- **Restituisce:** un intero.
- **Descrizione:** la funzione restituisce il punteggio associato alla tupla in ingresso.
- **Suggerimento:** Il calcolo del punteggio relativo ad una tupla (coordinate di un tiro) è effettuato in base alla distanza dal centro del bersaglio, come spiegato nella pagina introduttiva. Si utilizzi la funzione `calcola_distanza`.
- **Esempio:** Con riferimento alla figura riportata nella pagina introduttiva:
 - Il primo tiro di *Player 1* (il punto nero più vicino al centro) ha ottenuto un punteggio pari a 9. Gli altri tiri del giocatore hanno ottenuto 0, 3, 5, 5.
 - I tiri di *Player 2* hanno ottenuto un punteggio pari 2, 4, 3, 3, 4.

3. Definire la funzione `calcola_range_punteggi`

- **Parametri di ingresso:**
 - punteggi: lista di interi.
- **Restituisce:** un intero.
- **Descrizione:** la funzione restituisce la differenza tra valore massimo e minimo degli elementi maggiori di zero nella lista in ingresso. Se il numero di elementi maggiori di zero nella lista in ingresso è minore di due, la funzione restituisce 10 (valore massimo possibile di dispersione).
- **Esempio:** Con riferimento alla figura riportata nella pagina introduttiva:
 - Se in ingresso viene fornita la lista dei punteggi di *Player 1*, [9, 5, 5, 3, 0], la funzione restituisce 6.
 - Se in ingresso viene fornita la lista dei tiri di *Player 2*, [4, 4, 3, 2, 2], la funzione restituisce 2.

4. Infine, definire la funzione `main` in modo che svolga i seguenti compiti:

- crei la variabile `partita` come una lista di 10 tuple, ciascuna contenente due float nell'intervallo tra -1 e 1 ;
- calcoli il punteggio totale dei due giocatori, tenendo conto che i tiri dei giocatori sono memorizzati in `partita` in posizioni alternate;
- stampi l'esito della partita sulla base dei punteggi totali. Deve essere gestito anche l'esito di parità. Con riferimento all'esempio riportato nella pagina introduttiva, la stampa riporterà:

Vince Player1: 22 a 16