

## Test di Verifica AP-08-04-Test

1) La seguente porzione di programma C

```
enum esito {vittoria_interna, pareggio, vittoria_esterna};
typedef enum esito Esito;
typedef Esito Schedina[13];
Schedina mia_schedina;
mia_schedina[0] = pareggio;
mia_schedina[1] = vittoria_esterna;
mia_schedina[12] = vittoria_interna;
```

è corretta:

- V
- F

2) La seguente porzione di programma

```
enum esito {vittoriainterna, pareggio, vittoriaesterna};
typedef enum esito Esito;
typedef Esito Schedina[13];
Schedina miaschedina;
Esito risultato_Napoli_Milan;
risultato_Napoli_Milan = pareggio;
miaschedina[0]=pareggio;
miaschedina[1]=vittoria_esterna;
mia_schedina[12] = risultato_Napoli_Milan;
```

è corretta:

- V
- F

3) Si considerino i seguenti comandi al precompilatore:

```
#define AREA_RETTANGOLO BASE*ALTEZZA
#define BASE mia_base
#define ALTEZZA mia_altezza
```

e la porzione di main

```
int mia_area, mia_base, mia_altezza;
scanf("%d",&mia_base);
scanf("%d",&mia_altezza);
mia_area = AREA_RETTANGOLO;
```

```
printf("area: %d", mia_area);
```

allora l'istruzione `mia_area = AREA_RETTANGOLO;`  
viene trasformata in

- `mia_area = AREA_RETTANGOLO;`
- `mia_area = mia_base*mia_altezza;`
- `mia_area = BASE*ALTEZZA;`
- `AREA_RETTANGOLO = mia_base*mia_altezza;`

4) Si consideri la macro C `#define DIMEZZA(x) (x/2)`. Quali dei seguenti utilizzi della macro **non** consente di ottenere la corretta determinazione della divisione intera per 2 dell'argomento `numero` (dichiarato `int numero;`)?

- `risultato = DIMEZZA(2*numero);`
- `risultato = DIMEZZA(numero);`
- `risultato = DIMEZZA(2*numero+8);`
- `risultato = DIMEZZA((2*numero+8));`

5) Si consideri la macro C `#define DIMEZZA(x) ((x)/2)`. Allora tutti i seguenti utilizzi della macro consentono di ottenere la corretta determinazione della divisione intera per 2 dell'argomento `numero` (dichiarato `int numero;`):

```
risultato = DIMEZZA(2*numero);  
risultato = DIMEZZA(numero);  
risultato = DIMEZZA(2*numero+8);  
risultato = DIMEZZA((2*numero+8));
```

- V
- F

## Problemi aperti

1. Sviluppare una versione `double fattorialeD(int n)` della function per il fattoriale. Confrontando i valori del fattoriale calcolati da tale function e dalla function `float fattorialeF(int n)`, sviluppata nel modulo AP-06-06-C, determinare fino a quale valore di `n` la function `fattorialeF` è in grado di calcolare il valore corretto del fattoriale.
2. Scrivere una macro `EDISPARI(x)` che vale 1 se `x` è un numero (`int`) dispari e vale 0 se `x` è pari.
3. Scrivere una macro `EPARI(x)` che vale 1 se `x` è un numero (`int`) pari e vale 0 se `x` è dispari.
4. Scrivere una macro `ESPAZIO(x)` che vale 1 se `x` (di tipo `char`) è uno spazio e vale 0 altrimenti.