CATEGORIA A: QUIZ "COSA STAMPA"

A.1 ARITMETICA DEI PUNTATORI E TRONCAMENTO

Quiz A.1.1 - Troncamento Divisione (PRIMO COMPITINO 17/04/2024)

```c
int x = 2;
int *p = &x;
*p = (*p) / 4 * 2;
printf("%d", *p);
```

Opzioni: 0, 1, 2, 4 **Risposta: 0 Spiegazione:** Il puntatore p punta ad x=2, l'aritmetica del C esegue prima 2/4=0 (troncamento), poi 0*2=0

Quiz A.1.2 - Dangling Reference (PRIMO COMPITINO 17/04/2024)

```c
int *p;
{
    int y = 5;
    p = &y;
}
printf("%d", *p);
```

Opzioni: 5, 0, Errore di compilazione, Comportamento indefinito **Risposta: Comportamento indefinito Spiegazione:** Dangling reference - il puntatore p punta ad una variabile deallocata, quindi indefinita

Quiz A.1.3 - Matrici e Puntatori (PRIMO COMPITINO 17/04/2024)

```c
int m[3][4] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
printf("%d", (*(m+2))[1]);
```

Opzioni: 9, 10, 11, 6 **Risposta: 10 Spiegazione:** `*(m+2)` punta al primo elemento della terza riga (indice 2), `(*(m+2))[1]` si riferisce al secondo elemento della terza riga, ovvero 10

Quiz A.1.4 - Operatori di Incremento

```c
int i = 1;
printf("%d\n", 3*i+++3+i);
```

Opzioni: 7, 8, 9, 10 **Risposta: 8 Spiegazione:** `i++` usa il valore di i (1) poi lo incrementa. Quindi: 3*1+3+2 = 8

Quiz A.1.5 - Puntatori e Array Multidimensionali

```
int arr[2][3][4];
int *p = (int*)arr;
printf("%d", *(p+10));
```

Dato che `arr[1][0][2]` contiene il valore 42, cosa stampa? **Risposta: 42 Spiegazione:** `arr[1][0][2]` corrisponde a `*(p + 1*3*4 + 0*4 + 2)` = `*(p + 14)`. Ma se il quiz dice che `*(p+10)` corrisponde ad un elemento specifico, dobbiamo calcolare quale.

A.2 CONDIZIONI E LOGICA BOOLEANA

Quiz A.2.1 - Selezione Complessa (Q045)

```
int k1 = -1;
int k2 = 2;
int k3 = 0;
if(k1) {
    if(k3 || k2) {
        if(k3) {
            printf("A");
        }
    } else if (k3 && k2) {
        printf("B");
    }
} else {
    printf("C");
}
```

Opzioni: Niente, A, B, C **Risposta: Niente Spiegazione:** Conversione da intero a booleano: 0→Falso, altro→Vero. Le prime due condizioni sono vere ma non la terza. Al terzo if non è associato alcun else.

Quiz A.2.2 - Operatori Logici

```
int a = 5, b = 0, c = 3;
if (a && b || c) {
    printf("VERO");
} else {
    printf("FALSO");
}
```

Opzioni: VERO, FALSO **Risposta: VERO Spiegazione:** `(5 && 0) || 3` = `0 || 3` = `1` (vero)

Quiz A.2.3 - Short-Circuit Evaluation

```
int x = 5;
if (x < 3 && ++x > 4) {
    printf("%d", x);
} else {
    printf("%d", x);
}
```

Opzioni: 5, 6 **Risposta: 5 Spiegazione:** `x < 3` è falso, quindi `++x` non viene eseguito (short-circuit)

A.3 CICLI E ITERAZIONI

Quiz A.3.1 - Cicli Innestati

```
int n = 4, i, j;
for(j = 1; j <= n; j = j + 1) {
    for(i = 1; i <= n; i = i + 1) {
        printf("*");
    }
    printf("\n");
}
```

Domanda: Quanti asterischi stampa in totale? **Risposta: 16 Spiegazione:** 4 righe × 4 asterischi per riga = 16

Quiz A.3.2 - Ciclo con Condizione

```
int i = 0;
while (i < 5) {
    if (i % 2 == 0) {
        printf("%d ", i);
    }
    i++;
}
```

Cosa stampa: `0 2 4`

Quiz A.3.3 - For con Incremento Multiplo

```
for (int i = 0; i < 10; i += 3) {
    printf("%d ", i);
```

```
}
```

Cosa stampa: `0 3 6 9`

CATEGORIA B: ARITMETICA DEI PUNTATORI

B.1 MATRICI BIDIMENSIONALI

Quiz B.1.1 - Compitino 16/04/2025

Date le seguenti dichiarazioni:

```
double x[12][7];
float *y[7][15][9];
```

(a) Completare: `x[12][3]: *(x + ...)` **Risposta: Si accede oltre i limiti dell'array** (x ha solo righe 0-11)

(b) Completare: `y[6][8][2]: *(y + ...)` **Risposta:** `*(y + 6*15*9 + 8*9 + 2)` = `*(y + 810 + 72 + 2)` = `*(y + 884)`

Quiz B.1.2 - Matrici Standard

Data la dichiarazione:

```
int a[3][4];
int *p = &a[0][0];
```

Completare:

- `a[2][1]` → `*(p + ...)` **Risposta:** `*(p + 2*4 + 1)` = `*(p + 9)`
- `a[1][3]` → `*(p + ...)` **Risposta:** `*(p + 1*4 + 3)` = `*(p + 7)`

Quiz B.1.3 - Matrici Tridimensionali

Data la dichiarazione:

```
int c[3][5][4];
int *p = &c[0][0][0];
```

Completare: `c[2][1][3]` → `*(p + ...)` **Risposta:** `*(p + 2*5*4 + 1*4 + 3)` = `*(p + 40 + 4 + 3)` = `*(p + 47)`

Quiz B.1.4 - Errori Comuni

```
char b[3][2];
char *p = &b[0][0];
```

Cosa rappresenta: `b[1][2] → *(p + ...)`? **Risposta:** `*(p + 1*2 + 2)` = `*(p + 4)` - **ERRORE**: l'array ha solo 2 colonne (indici 0-1)

B.2 CONVERSIONI INDICE-POSIZIONE

Quiz B.2.1 - Array come Matrice

```
int array[12] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

Se consideriamo l'array come matrice 3×4:

- Elemento in posizione 7 (valore 8) corrisponde a quale riga/colonna?

Risposta:

- Riga: `7/4 = 1`
- Colonna: `7%4 = 3`
- Quindi: `matrice[1][3]`

Quiz B.2.2 - Formula Generale

Per una matrice `m×n`, l'elemento in posizione `i` corrisponde a:

- **Riga:** `i/n`
- **Colonna:** `i%n`

CATEGORIA C: DICHIARAZIONI COMPLESSE

C.1 PUNTATORI E ARRAY

Quiz C.1.1 - Primo Compitino (17/04/2024)

Scrivere le dichiarazioni per:

1. `x` è un array di 5 puntatori a carattere **Risposta:** `char *x[5];`
2. `p` è un puntatore ad un array di 7 puntatori a double **Risposta:** `double* (*p)[7];`

Quiz C.1.2 - Compitino 16/04/2025

`p` è un puntatore ad un vettore di 5 elementi, ciascuno dei quali è un puntatore a carattere

Risposta: `char * (*p)[5];`

Quiz C.1.3 - Dichiarazioni Avanzate

Scrivere le dichiarazioni per:

1. `q` è un puntatore a funzione che restituisce int e prende due parametri int **Risposta:** `int (*q)(int, int);`
2. `r` è un array di 10 puntatori a funzioni che restituiscono void e prendono un char ***Risposta:*** *`void (*r[10])(char*);`*
3. `s` è un puntatore ad un array di 3 puntatori a int **Risposta:** `int *(*s)[3];`

Quiz C.1.4 - Secondo Appello P1

"Definire un puntatore ad un array di puntatori ad intero"

Se l'array ha 8 elementi: **Risposta:** `int *(*ptr)[8];`

C.2 INTERPRETAZIONE DICHIARAZIONI

Quiz C.2.1 - Analisi Tipo

Data la dichiarazione: `int *p[5][3];`

Cosa rappresenta? **Risposta:** `p` è una matrice 5×3 di puntatori ad int

Quiz C.2.2 - Precedenza Operatori

Data: `int (*p[5])[3];`

Cosa rappresenta? **Risposta:** `p` è un array di 5 puntatori a array di 3 int

---

CATEGORIA D: ANALISI FUNZIONI RICORSIVE

D.1 FUNZIONI MATEMATICHE

Quiz D.1.1 - Potenza di 2 (Primo Compitino)

```
int funzione(int n) {
    if (n == 1)
        return 2;
    else
```

```
        return 2 * funzione(n-1);
}
```

PRE: n >= 1 **POST:** restituisce 2^n **Cosa fa:** Calcola potenze di 2

Quiz D.1.2 - Massimo Array (Compitini 2022)

```
int f(int X[], int dim) {
    if (dim == 1)
        return X[0];
    else {
        int m = f(X+1, dim-1);
        return (X[0] > m) ? X[0] : m;
    }
}
```

PRE: dim > 0, X contiene almeno dim elementi **POST:** restituisce il valore massimo in X

Quiz D.1.3 - Prodotto Fattoriale (Compitini 2022)

```
int f(int n, int m) {
    if (n == m)
        return n;
    else
        return n * f(n-1, m);
}
```

PRE: n ≥ m ≥ 1 **POST:** Calcola n × (n-1) × (n-2) × ... × m

Quiz D.1.4 - Array Tutti Pari (Esame 29/8/22)

```
int f(int X[], int dim) {
    if (dim == 0)
        return 1;
    if (X[0] % 2 == 0)
        return f(X+1, dim-1);
    else
        return 0;
}
```

PRE: dim ≥ 0, X contiene almeno dim elementi **POST:** Restituisce 1 se tutti gli elementi sono pari, 0 altrimenti

D.2 CORRETTEZZA FUNZIONI

Quiz D.2.1 - Dimostrazione Induttiva

Per la funzione massimo array, dimostrare la correttezza:

Caso base: Se dim == 1, il singolo elemento è il massimo ✓ **Caso induttivo:** Per ipotesi induttiva, f(X+1, dim-1) restituisce il massimo del resto. Confrontando con X[0] otteniamo il massimo globale ✓

CATEGORIA E: QUIZ LOGICI E ALGORITMI

E.1 TABELLE DI VERITÀ

Quiz E.1.1 - Esco o Sto a Casa

```
int piove = 0;
int ombrello = 1;
if(!piove) {
    printf("esco");
} else if (ombrello) {
    printf("esco");
} else {
    printf("sto a casa");
}
```

Tabella di Verità Completa:

piove	ombrello	stampa esco
F	F	V
F	V	V
V	F	F
V	V	V

Quiz E.1.2 - Versione Semplificata

Come riscrivere il codice precedente?

```
if(piove && !ombrello) {
    printf("sto a casa");
} else {
    printf("esco");
}
```

E.2 ALGORITMI SU MATRICI

Quiz E.2.1 - Mosse Alfiere

Data una scacchiera 8×8, quante caselle può raggiungere un alfiere dalla posizione (3,3)?

Risposta: 13 caselle (4 direzioni diagonali)

Quiz E.2.2 - Percorsi su Griglia

In una griglia 3×3, quanti percorsi diversi esistono dall'angolo in alto a sinistra a quello in basso a destra (solo mosse destra/giù)?

Risposta: 6 percorsi

CATEGORIA F: QUIZ AVANZATI E PATTERN COMPLESSI

F.1 GESTIONE MEMORIA

Quiz F.1.1 - Memory Leak

```
void funzione() {
    int *p = malloc(sizeof(int) * 10);
    *p = 42;
    printf("%d", *p);
    // Cosa manca?
}
```

Risposta: Manca `free(p);`

Quiz F.1.2 - Dangling Pointer

```
int* crea_array() {
    int arr[5] = {1,2,3,4,5};
    return arr;  // ERRORE!
}
```

Problema: Restituisce puntatore a variabile locale **Soluzione:** Usare malloc o dichiarare static

F.2 OTTIMIZZAZIONE ALGORITMI

Quiz F.2.1 - Fibonacci Efficiente

Quale implementazione è più efficiente per calcolare F(40)?

1. Ricorsiva naive: O(2^n)
2. Iterativa: O(n)
3. Memoization: O(n)

Risposta: 2 o 3 (iterativa o memoization)

Quiz F.2.2 - Ricerca Binaria

In un array ordinato di 1000 elementi, quanti confronti servono al massimo per trovare un elemento?

Risposta: $\lceil \log_2(1000) \rceil = 10$ confronti

F.3 STRUTTURE DATI

Quiz F.3.1 - Lista vs Array

Quale operazione è più efficiente in una lista collegata rispetto ad un array?

1. Accesso casuale
2. Inserimento in testa
3. Ricerca lineare
4. Accesso sequenziale

Risposta: 2 (Inserimento in testa) - O(1) vs O(n)

Quiz F.3.2 - BST vs Lista

In un BST bilanciato con 1000 nodi, quanti confronti servono per la ricerca?

Risposta: O(log n) = circa 10 confronti

CATEGORIA G: QUIZ SU SINTASSI E COMPILAZIONE

G.1 ERRORI DI SINTASSI

Quiz G.1.1 - Sintassi Primo Programma (Q042)

"Ogni programma deve avere una funzione di nome 'main'"

Risposta: Vero

Quiz G.1.2 - Dichiarazioni Errate

Quale dichiarazione è corretta?

1. `int arr[];`
2. `int arr[10];`
3. `int arr[3.5];`
4. `int arr[-1];`

Risposta: 2

Quiz G.1.3 - Include Guard

```
#ifndef HEADER_H
#define HEADER_H
// contenuto header
#endif
```

Scopo: Evitare inclusioni multiple

G.2 PREPROCESSING

Quiz G.2.1 - Macro

```
#define MAX(a,b) ((a) > (b) ? (a) : (b))
int x = 5, y = 3;
printf("%d", MAX(++x, ++y));
```

Problema: Incremento multiplo **Risultato:** Comportamento indefinito

Quiz G.2.2 - Conditional Compilation

```
#ifdef DEBUG
    printf("Debug: x = %d\n", x);
#endif
```

Quando viene compilato: Solo se DEBUG è definito

CATEGORIA H: QUIZ PRATICI E CASI REALI

H.1 ALGORITMI ORDINAMENTO

Quiz H.1.1 - Bubble Sort

Per ordinare un array di 5 elementi, quanti confronti fa bubble sort nel caso peggiore?

Risposta: n(n-1)/2 = 5×4/2 = 10 confronti

Quiz H.1.2 - Merge Sort

Qual è la complessità temporale di merge sort?

Risposta: O(n log n) in tutti i casi

H.2 PROBLEMI GEOMETRICI

Quiz H.2.1 - Punto nel Rettangolo

Un punto (5,3) è interno al rettangolo con vertici (2,1) e (8,6)?

Risposta: Sì (2 < 5 < 8 e 1 < 3 < 6)

Quiz H.2.2 - Distanza Euclidea

```
double distanza(int x1, int y1, int x2, int y2) {
    return sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
}
```

Distanza tra (0,0) e (3,4)? **Risposta:** 5.0

H.3 PROBLEMI DI CONTEGGIO

Quiz H.3.1 - Fattoriale

```
int fattoriale(int n) {
    return (n <= 1) ? 1 : n * fattoriale(n-1);
}
```

Valore di fattoriale(5)? **Risposta:** 120

Quiz H.3.2 - Somma Array

```
int somma(int arr[], int n) {
    return (n == 0) ? 0 : arr[0] + somma(arr+1, n-1);
}
```

Per arr={1,2,3,4,5}, qual è somma(arr,5)? **Risposta:** 15

STRATEGIE DI RISOLUZIONE

Per Quiz "Cosa Stampa"

1. **Seguire l'ordine di esecuzione** riga per riga
2. **Attenzione alle precedenze** degli operatori
3. **Considerare i troncamenti** nelle divisioni tra interi
4. **Verificare la validità** dei puntatori
5. **Short-circuit evaluation** per operatori logici

Per Aritmetica dei Puntatori

1. **Identificare le dimensioni** degli array
2. **Applicare la formula:** `*(base + offset)`
3. **Calcolare l'offset:** `riga*cols + col` per 2D
4. **Controllare i limiti** dell'array
5. **Formula 3D:** `piano*righe*cols + riga*cols + col`

Per Dichiarazioni Complesse

1. **Leggere da destra a sinistra**
2. **Seguire la precedenza:** `[]` prima di `*`
3. **Usare parentesi** per forzare l'ordine
4. **Verificare con esempi** pratici
5. **Ricordare:** `*p[n]` ≠ `(*p)[n]`

Per Analisi Ricorsiva

1. **Identificare caso base** e verifica terminazione
2. **Tracciare alcune chiamate** per piccoli valori
3. **Verificare la riduzione** del problema
4. **Scrivere PRE e POST** condizioni
5. **Dimostrare correttezza** per induzione

Per Quiz Logici

1. **Costruire tabella di verità** se necessario
2. **Applicare leggi di De Morgan**
3. **Considerare short-circuit**
4. **Conversioni implicite** 0→Falso, altro→Vero
5. **Precedenza operatori:** `!` > `&&` > `||`

ERRORI COMUNI DA EVITARE

1. Aritmetica dei Puntatori

- Confondere `p[i][j]` con `*(p + i + j)`
- Dimenticare le parentesi: `*(p+i*cols+j)`
- Non controllare i limiti degli array

2. Dichiarazioni

- Confondere `int *p[5]` con `int (*p)[5]`
- Dimenticare le parentesi nelle dichiarazioni complesse

3. Ricorsione

- Caso base irraggiungibile
- Stack overflow per input grandi
- Non verificare le precondizioni

4. Logica Booleana

- Dimenticare la precedenza degli operatori
- Non considerare il short-circuit
- Confondere `=` con `==`

5. Gestione Memoria

- Memory leak (dimenticare free)
- Dangling pointer
- Double free

Quiz raccolti dagli esami 2022-2025 con pattern aggiuntivi per preparazione completa