

1. 3 points Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int n = 0xA - 6, x, y;
3     for(x = 1; x <= n; x++){
4         for(y = n; y >= x; y--){
5             printf("%c", 'A'-1 + x);
6         }
7     }
8     printf("\n");
9 }
10 }
```

AAAA
BBB
CC
D

2. 3 points Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int i, j;
3     int m, m = 4;
4     for (i = 1; i <= n; i++){
5         for (j = 1; j <= m; j++){
6             (i==1 || i==n || j==1 || j==m)\
7             ? printf("* ") : printf("- ");
8         }
9         printf("\n");
10    }
11 }
```

* * * *
* _ _ *
* _ _ *
* * * *

3. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int i;
3     i = 0;
4     if(i = 15, 10, 5)
5         printf("%d Mele", i);
6     else
7         printf("%d Pere", i);
8 }
```

15 Mele

4. 2 points Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int a;
3     a = 10;
4     do{
5         while(a++ < 10);
6     }while(a++ <= 11);
7     printf("%d", a);
8 }
```

a = 14

5. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int i;
3     for(i = 1; i++ <= 1; i++){
4         i++;
5     }
6     printf("%d", i);
7 }
```

i = 5

6. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int i, j, *ptr, *ptr1;
3     i = 10;
4     j = 10;
5     ptr = &i;
6     ptr1 = &j;
7     if(ptr == ptr1)
8         printf("True");
9     else
10        printf("False");
11 }
```

Output = False

7. 2 points Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int *p = (int*) 64;
3     int *q = (int*) 40;
4     printf("%d", p-q);
5     return 0;
6 }
```

Output = 6

Proof:
 $64 - 40 = 24$
 $24 / \text{sizeof(int)} = 6$

8. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int p , a = 71;
3     p = &a;
4     *(int*)p = 8;
5     printf("%d", a);
6     return 0;
7 }
```

Output = Segmentation fault

9. 3 points Scrivere cosa stampa il seguente codice.

```
1 struct Data
2 {
3     short a;
4     int = b;
5     char = c;
6 }d;
7
8 int main(){
9     printf("%ld", sizeof(struct Data));
10 }
```

Output:12

Proof:

short: 2 bytes, 2 padding bytes

int: 4 bytes

char: 1 bytes, 3 padding bytes

10. 2 points Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     unsigned int a = -10;
3     int b = ~9;
4     int result;
5     if(b==a)
6         printf("equal");
7     else
8         printf("unequal");
9     return 0;
10 }
```

Output = equal

11. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int a, b, c;
3     a = b = c = 100;
4     if((a == b == c) > 1)
5         printf("True\n");
6     else
7         printf("False\n");
8     return 0;
9 }
```

Output = False

12. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     float arr[] = {12.4, 2.3, 4.5, 6.7};
3     printf("%d", sizeof(arr)/sizeof(arr[0]));
4     return 0;
5 }
```

Output = 4

13. 2 points Trovare il valore finale di "result" ed "y".

```
1 int main(){
2     char result = 125;
3     {int y = 012;}
4     result += 5;
5     printf("%d", result);
6     printf("%d", y);
7     return 0;
8 }
```

Output:

result = -126

y = variabile non definita

14. 2 points Elencare tutte le conversioni di tipo presenti.

```
1 int main(){
2     char d = 'A';
3     unsigned int a = -3LL;
4     long int b = 2 * 'f';
5     const int c = 21.9;
6     c = (int)d;
7     unsigned short e = ~9;
8     return 0;
9 }
```

-3LL da long long ad unsigned int
'f' da char ad int
2 * 'f' da int a long int
21.9 da float ad int
9 da int ad unsigned short

15. 1 point Scrivere cosa stampa il seguente codice.

```
1 int main(){
2     int arr[][3] = {6, 5, 4, 3, 2, 1};
3     printf("%d %d", arr[0][0], arr[1][0]);
4     return 0;
5 }
```

Output = 6 3