

1. Evaluate the following conversions.

- a. $(200)_{\text{dec}} = (11001000)_{\text{bin}}$
- b. $(111111100101010)_{\text{bin}} = (7F2A)_{\text{hex}}$
- c. $(6C)_{\text{hex}} = (108)_{\text{dec}}$
- d. $(52746)_{\text{oct}} = (55E6)_{\text{hex}}$

2. Mark valid names with x. (Do not mark invalid ones)

- | | |
|--|---|
| ☒ LONG | ☐ X++ |
| ☒ printf | ☐ %count% |
| ☐ 10_July | ☐ i= |
| ☐ May.1 | ☒ U_1_2_3_4_5 |
| ☒ int_16 | ☒ A1EC23hex |

3. Determine what is printed on the screen when the given code is executed.

- a. `int A=2,B=2;
A+=(++B + B++);
printf("%d %d",A,B);`

8 4

- c. `int A=3;
A/=2; A*=2;
printf("%d",--A);`

1

- b. `float B=1.5,A=1.5;
int C=1;
B+=C+=A;
printf("B=%4.2f",B);`

B=3.50

- d. `double B,A; B=A=3.14;
A=2.0*((int)B/2.0);
if(A<=B)
printf("%3.1lf",B);
printf("%d", (int)A);`

3.13

4. Modify the following multiplication and division calculator in order to handle "division by zero" exception. Indicate the line number you need to modify too.

```
1. int main(void){
2.     int A,B; char op;
3.     printf("Enter A@B : ");
4.     scanf("%d %c %d", &A, &op, &B);
5.     if(op=='*')
6.         printf("%d\n",A*B);
7.     else if(op=='/')
8.         printf("%d\n",A/B);
9.     else
10.        printf("unknown op. %c\n",op);
11.    return 0;
12.}
```

line 8 should be changed to read

```
if(B==0)
    printf("Error.\n");
else
    printf("%d\n",A/B);
```