Machine Problem 1 – Loops and Arrays

1.

Code:

```c
#include <stdio.h>

int main(void){
    int i;

    i=1;
    while(i<=128){
        printf("%d ", i);
        i*=2;
    }

    return 0;
}
```

Output:

```
\C>  & 'c:\Users\A\.vscode\extensions\ms-vscode.cpptools-1.14.5-
' '--stdout=Microsoft-MIEngine-Out-logsjg5y.u43' '--stderr=Micro
mingw64\bin\gdb.exe' '--interpreter=mi'
1 2 4 8 16 32 64 128
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

2.

Assuming that the initial value of i in our program is 11, letter c is the one that is not equivalent to the other two loops. Given that i=1, a and b will not run, while c would run at least once.

a.

Code:

```c
int i=11;

//a.
while(i<10){
    printf("C is great %d\n", i+1);
    i+=1;
}
```

Output:

```
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

b.

Code:

```c
for(; i<10;){
    printf("C is great %d\n", i+1);
    i+=1;
}
```

Output:

```
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

c.

Code:

```c
do{
    printf("C is great %d\n", i+1);
    i+=1;
}while(i<10);
```

Output:

```
C is great 12
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

3.

Code:

```c
#include <stdio.h>

int main(void){
    int i, product=1;

    // Initial value of product is 1 and it is multiplied to 2 and updated
    // for each iteration as long as i is less than or equal to 7
    for(i=0; i<=7; i++){
        printf("%d ", product);
        product*=2;
    }
}
```

Output:

```
1 2 4 8 16 32 64 128
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

4.

Code:

```c
#include <stdio.h>

int main(void){
    int i, product=1;

    printf("TABLE OF POWERS OF TWO\nn\t2 to the n\n\n");

    // Initial value of product is 1 and it is multiplied to 2 and updated
    // for each iteration as long as i is less than or equal to 10. Value of
    // product is tabbed printed to the right of the value of i.
    for(i=0; i<=10; i++){
        printf("%d\t%d\n", i, product);
        product*=2;
    }

}
```

Output:

```
TABLE OF POWERS OF TWO
n       2 to the n

0       1
1       2
2       4
3       8
4       16
5       32
6       64
7       128
8       256
9       512
10      1024
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

5.

Code:

```c
#include <stdio.h>

int main(void){
    int startDay, daysOfMonth, i;

    printf("Enter number of days in month: ");
    scanf("%d", &daysOfMonth);

    printf("Enter starting day of the week (1=Sun, 7=Sat): ");
    scanf("%d", &startDay);

    printf("\n");

    //If daysOfMonth entered is less than 28, more than 31, negative, or startDay is negative tell user that input is invalid.
    if(((daysOfMonth<28 || daysOfMonth>31) || daysOfMonth<=0) || (startDay<=0)){
        printf("Invalid input. Try again");
    }

    else{
        //Prints spaces depending to the starting day of the month provided by the user.
        //For example starting day is 3 therefore print spaces for the first 2 spots.
        for(i=1; i<startDay; i++){
            printf("    ");
        }

        //After printing the spaces print the days of the month
        //%2d is used since days can be two digits
        //If the day of the week reaches 7 print on the next line.
        for(i=1; i<=daysOfMonth; i++){
            printf("%2d ", i);
            if((i+startDay-1)%7==0){
                printf("\n");
            }
        }
    }
}
```

Output:

```
Enter number of days in month: 31
Enter starting day of the week (1=Sun, 7=Sat): 3

       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
PS C:\Users\A\Desktop\School Works\FIRST YEAR PROGRAMMING\C>
```

6.

a.

Code:

```
bool pathway[8] = {true, [2]=true};
```

Output:

```
1 0 1 0 0 0 0 0
```

b.

Code:

```
bool pathway[8] = {true, false, true};
```

Output:

```
1 0 1 0 0 0 0 0
```

7.

a.

```
bool road_networks[8][8] = {
[0][0]=true, [1][0]=true, [5][0]=true, [6][0]=true,
[0][1]=true, [1][1]=true, [2][1]=true, [1][2]=true,
[2][2]=true, [5][2]=true, [3][3]=true, [4][3]=true,
[6][3]=true, [2][4]=true, [3][4]=true, [4][4]=true,
[0][5]=true, [2][5]=true, [5][5]=true, [7][5]=true,
[6][6]=true, [7][7]=true
};
```

b.

```
 a      b      c      d      e      f      g      h
 1      1     [0]    [0]     0      1      0      0
 1      1     [1]    [0]     0      0      0      0
[0]    [1]    [1]    [0]    [1]    [1]    [0]    [0]
[0]    [0]    [0]    [1]    [1]    [0]    [0]    [0]
 0      0     [0]    [1]     1      0      0      0
 1      0     [1]    [0]     0      1      0      0
 1      0     [0]    [1]     0      0      1      0
 0      0     [0]    [0]     0      1      0      1
```

c.

Code:

```
do{
    printf("Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): ");
    scanf("%d", &response);

    if ((response==0 || response==1) || (response==5 || response==7)){
        printf("Nearest charging station: C\n");
    }
    else if((response==4) || (response==6)){
        printf("Nearest charging station: D\n");
    }
    else if(response==2 || response==3){
        printf("You are already in a charging station.\n");
    }
    else{
        printf("Invalid input. Try again\n");
    }

    printf("Continue?(Yes=1 No=0) ");
    scanf("%d", &i);
}while(i!=0);
```

Output:

```
Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): 5
Nearest charging station: C
Continue?(Yes=1 No=0) 1
Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): 6
Nearest charging station: D
Continue?(Yes=1 No=0) 1
Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): 0
Nearest charging station: C
Continue?(Yes=1 No=0) 1
Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): 3
You are already in a charging station.
Continue?(Yes=1 No=0) 1
Location(a=0 b=1 c=2 d=3 e=4 f=5 g=6 h=7): 89
Invalid input. Try again
Continue?(Yes=1 No=0) 0
```

d.

Code:

```
#define arraySize ((int)sizeof(road_networks)/(int)sizeof(road_networks[0][0]))
```

Output:

```
Array size is: 64
```