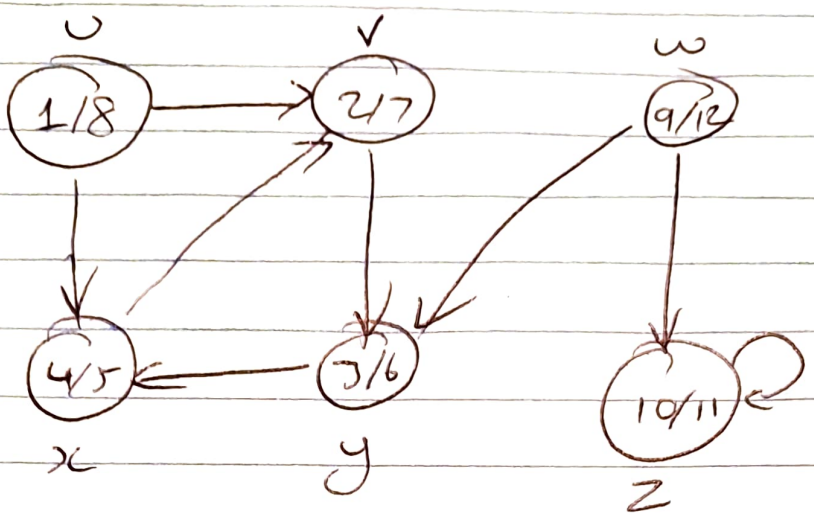
Qu 1 a



Qu1b

When using srand use the unassigned int %u. ~~at the end of program~~

Use <time.h> to generate the srand %u which will generate the random number based on the time.

This should stop the number generation from repeating.

Q.2

```
#include <stdio.h>

int main() {
    int km, m;

    printf("Enter distance in kilometres: ");
    scanf_s("%d", &km);

    printf("Enter distance in metres: ");
    scanf_s("%d", &m);

    int totalD = km * 1000 + m;
    printf("Total distance travelled: %od metres\n",
     totalD);

    return 0;
}
```

Qu 3

in both lines with "void A1Function(const int b[7]);" the parameter "const" doesn't allow C to modify the variable. removing "const" from both lines fixes the program.

Qu 4

```
#include <stdio.h>
float Flagest (float num1, float num2, float num3,
  float num4, float num5) {
  float largest = num1;
  if (num2 > largest) {
    largest = num2; }
  if (num3 > largest) {
    largest = num3; }
  if (num4 > largest) {
    largest = num4); }
  if (num5 > largest) {
    largest = num5}; }
  return largest;
}

int main () {
  float num1, num2, num3, num4, num5;
  printf ("Enter 1st number: ");
  scanf_s ("%f", &num1);
  printf ("Enter 2nd number: ");
  Scanf_s ("%f", &num2);
  printf (Enter 3rd number: ");
  Scanf_s ("%f", &num3);
```

```
printf ("Enter 4th number: ");
scanf_s("%f", num4);
printf ("Enter 5th number: ");
scanf_s ("%f", num5);

float largest = Flargest (num1, num2, num3, num4,
    num5);

printf ("largest number is: %.2\n", largest);
return 0
}
```

Q.5

```
#include <stdio.h>
int main () {
    int num1, num2;
    int *ptr1, *ptr2;
printf ("Enter 1st int: ");
scanf_s ("%d", &num1);
printf ("Enter 2nd int: ");
scanf_s ("%d", &num2);
ptr1 = num1
ptr2 = num2
int result = *ptr1 - *ptr2;
printf ("%d minus %d is %d", *ptr2, *ptr1, result);
return 0;
}
```

Q6 (a) Bubble sort compares 2 integers sequentially. I.e. it will compare the first pair (first and second numbers) then order them by size. It then compares the second and third numbers and orders them by sizes. Once the process completes through the whole array it will begin again.
When the process completes with no swaps the program will stop.

(b) 10 comparisons.

Q7

```
#include <stdio.h>
int main () {
    int mirrors [] = {0, 2, 1, 3};
    int length = sizeof (mirrors) / sizeof (mirrors[0]);
    for (int i=0; i<length; i++) {
        if (mirrors[i] == i) {
            printf ("%d\n", mirrors[i]);
        }
    return 0;
}
```