Control statements –II

Implement C programs for the following problem statements:

1. Program to print words corresponding numbers below 9. (Switch)

```
#include <stdio.h>
int main()
{
   int n, num = 0;
   printf("Enter any number less than 9: ");
   scanf("%d", &n);

   while(n != 0)
   {
      num = (num * 10) + (n % 10);
      n /= 10;
   }

   while(num != 0)
   {
      switch(num % 10)
      {
         case 0:
```

```
        printf("Zero ");
        break;
    case 1:
        printf("One ");
        break;
    case 2:
        printf("Two ");
        break;
    case 3:
        printf("Three ");
        break;
    case 4:
        printf("Four ");
        break;
    case 5:
        printf("Five ");
        break;
    case 6:
        printf("Six ");
        break;
    case 7:
        printf("Seven ");
        break;
```

Samuela Abigail
71762108039

```
            case 8:
                printf("Eight ");
                break;
        }

        num = num / 10;
    }

    return 0;
}
```

Inference- since we are asked to print numbers below 9, it can print from zero to eight only.

2. Program to calculate Arithmetic Operations depending on operator.

```
#include <stdio.h>
int main() {
  double a, b;
  char op;

  printf("Enter character of arithmetic operation (+, -, *, /): ");
  scanf("%c", &op);

  printf("Enter value of 2 numbers: ");
  scanf("%lf %lf", &a, &b);

  switch (op) {
```

```
    case '+':
    printf("Addition of %lf and %lf is %lf", a, b, a+b);
     break;

    case '-':
    printf("Subtraction of %lf and %lf is %lf", a, b, a-b);
     break;

    case '*':
     printf("Multiplication of %lf and %lf is %lf", a, b,a*b);
     break;

    case '/':
    printf("Division of %lf and %lf is %lf", a, b, a/b);
     break;

    default:
     printf("Error! operator is not correct");
  }

   return 0;
}
```

Inference- switch case statement is used to code for a program that calculates in accordance to the arithmetic operator.

3. Display numbers in the following format.

1

1 2

1 2 3

Samuela Abigail
71762108039

1 2 3 4

1 2 3 4 5

```
#include <stdio.h>
int main() {
  int i, j, rows;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
  for (i = 1; i <= rows; ++i) {
    for (j = 1; j <= i; ++j) {
      printf("%d ", j);
    }
    printf("\n");
  }
  return 0;
}
```

Inference- the number pyramid is printed based on increment operand.

4. Display numbers in the following format.

7 5 1 4 2

5 1 8 4

2 7 5

Samuela Abigail
71762108039

3 2

```
#include <stdio.h>
int main() {
   int i, j, k, l;
   int row1[5]= {7, 5, 1, 4, 2};
   int row2[4]= {5, 1, 8, 4};
   int row3[3]= {2, 7, 5};
   int row4[2]= {3, 2};

   for(i=0; i<5; i++)
   {
      printf("%d ", row1[i]);
   }

  printf("\n");

for(j=0; j<4; j++)
   {
      printf("%d ", row2[j]);
   }

     printf("\n");
```

Samuela Abigail
71762108039

```
  for(k=0; k <3; k++)
  {
    printf("%d ", row3[k]);
  }

   printf("\n");

  for(l=0; l <2; l++)
  {
    printf("%d ", row4[l]);
  }
  return 0;

}
```

Inference- Arrays and for loop is used for this pyramid since the numbers making up the pyramid have no sequence.

5. Display Pascal's Triangle. Enter the number of rows: 6

```
#include <stdio.h>
int main() {
  int vline, coeff = 1, space, i, j;
  printf("Enter the number of rows (vertical lines): ");
  scanf("%d", &vline);
```

```
  for (i = 0; i < vline; i++) {
    for (space = 1; space <= vline - i; space++)
      printf("  ");
    for (j = 0; j <= i; j++) {
      if (j == 0 || i == 0)
        coeff = 1;
      else
        coeff = coeff * (i - j + 1) / j;
      printf("%4d", coeff);
    }
    printf("\n");
  }
  return 0;
}
```

Inference- the variable called space is used to arrange the numbers in a pyramid-like equilateral triangle instead of a normal triangle. Variable vline tells number of rows in Pascal triangle to be printed. Variables i and j code for constituent numbers in rows and columns of Pascal triangle respectively.

Samuela Abigail
71762108039