Q1)

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#define TRUE    1
#define FALSE   0
#define NUM_ROWS 25

typedef struct
{
    char name[100];
    double n;
    double slope;
    double width;
    double maxDepth;
}CHANNEL;

void getInput(CHANNEL *);
void getPositiveValue(CHANNEL *);
void fillArray(CHANNEL *, double *, double *);
double computeVelocity(CHANNEL , double *);
void displayTable (CHANNEL , double *, double *);


int main()
{
    CHANNEL ch;
    double depth[25];
    double vel[25];

    getInput(&ch);

    fillArray(&ch, depth, vel);

    displayTable (ch, depth, vel);
    return 0;
}
```

```c
void getInput(CHANNEL *ch)
{
    printf("Give the name of the channel: ");
    fgets(ch->name, 100, stdin);
    ch->name[strcspn(ch->name, "\n")] = '\0';

    getPositiveValue(ch);
}

void getPositiveValue(CHANNEL *ch)
{
    int i, flag;
    double array[4] = {ch->n,ch->slope,ch->width,ch->maxDepth,};
    char variables[4][30] = {"coeffecient of roughness", "slope", "channel width", "maximum depth of the channel"};
    for(i = 0;i <= 3; i++)
    {
        do
        {
            flag = TRUE;
            printf("Give the %s: ", variables + i);
            scanf("%lf",&array[i]);

            if(array[i] <= 0)
            {
                printf("value must be greater than 0.\n");
                flag = FALSE;
            }
        } while(flag == FALSE);
    }

    ch->n = array[0];
    ch->slope = array[1];
    ch->width = array[2];
    ch->maxDepth= array[3];
}
```

```c
void fillArray(CHANNEL *ch, double *depth, double *vel)
{
    double increment;
    int i;
    increment = (ch->maxDepth)/(NUM_ROWS);
    for (i = 0; i<NUM_ROWS; i++)
    {
        depth[i]= increment*(i+1);
        vel[i]= computeVelocity(*ch, &depth[i]);
    }
}

double computeVelocity(CHANNEL ch, double *depth)
{
    double velocity;
    velocity = ((sqrt(ch.slope))/ch.n);
    double var = (ch.width/(*depth))/(ch.width+(2*(*depth)));
    velocity = velocity *pow(var,(2.0/3.0));
    return velocity;
}

void displayTable(CHANNEL ch, double *depth, double *vel)
{
    // Complete the function - call the function computeVelocity
    int i;
    printf("\nChannel data for \"%s\"\n\tCoefficient of roughness: %g\n\tSlope: %g\n\tWidth: %g\n\tMaximum depth:%g\n  ", ch.name,ch.n,ch.slope,ch.width,ch.maxDepth);

    printf("\t%s\t\t%s\n", "Depth", "Velocity");
    printf("-----------------------------------------\n");

    for (i =0 ; i<NUM_ROWS ; i++)
    {
        printf("\t%.4lf\t\t%.6lf\n", depth[i], vel[i]);
    }
}
```

Test case 1

```
Give the name of the channel: Channel 1
Give the coeffecient of roughness: 0.035
Give the slope: 0.0001
Give the channel width: 10
Give the maximum depth of the channel: 4.2

Channel data for "Channel 1"
        Coefficient of roughness: 0.035
        Slope: 0.0001
        Width: 10
        Maximum depth:4.2
        Depth           Velocity
----------------------------------------
        0.1680          0.917961
        0.3360          0.566077
        0.5040          0.423161
        0.6720          0.342380
        0.8400          0.289368
        1.0080          0.251450
        1.1760          0.222759
        1.3440          0.200172
        1.5120          0.181859
        1.6800          0.166669
        1.8480          0.153840
        2.0160          0.142843
        2.1840          0.133301
        2.3520          0.124935
        2.5200          0.117535
        2.6880          0.110939
        2.8560          0.105020
        3.0240          0.099677
        3.1920          0.094829
        3.3600          0.090410
        3.5280          0.086363
        3.6960          0.082644
        3.8640          0.079214
        4.0320          0.076040
        4.2000          0.073095

Process returned 0 (0x0)   execution time : 17.949 s
Press any key to continue.
```

Test Case 2

```
Give the name of the channel: Channel 2
Give the coeffecient of roughness: 0.0013
Give the slope: 0.0032
Give the channel width: 2
Give the maximum depth of the channel: 11.5

Channel data for "Channel 2"
        Coefficient of roughness: 0.0013
        Slope: 0.0032
        Width: 2
        Maximum depth:11.5
        Depth           Velocity
----------------------------------------
        0.4600          56.740241
        0.9200          29.778691
        1.3800          19.693713
        1.8400          14.450211
        2.3000          11.266883
        2.7600          9.146052
        3.2200          7.641631
        3.6800          6.524830
        4.1400          5.666601
        4.6000          4.988857
        5.0600          4.441703
        5.5200          3.991847
        5.9800          3.616268
        6.4400          3.298570
        6.9000          3.026780
        7.3600          2.791958
        7.8200          2.587308
        8.2800          2.407573
        8.7400          2.248630
        9.2000          2.107203
        9.6600          1.980655
        10.1200         1.866844
        10.5800         1.764015
        11.0400         1.670711
        11.5000         1.585720

Process returned 0 (0x0)   execution time : 17.404 s
Press any key to continue.
```

Test case 3

```
Give the name of the channel: Channel 3
Give the coeffecient of roughness: 0.17
Give the slope: 0.041
Give the channel width: 40
Give the maximum depth of the channel: 1.5

Channel data for "Channel 3"
        Coefficient of roughness: 0.17
        Slope: 0.041
        Width: 40
        Maximum depth:1.5
        Depth           Velocity
-----------------------------------------
        0.0600          7.756067
        0.1200          4.876297
        0.1800          3.713932
        0.2400          3.059721
        0.3000          2.631589
        0.3600          2.325820
        0.4200          2.094561
        0.4800          1.912415
        0.5400          1.764548
        0.6000          1.641663
        0.6600          1.537612
        0.7200          1.448154
        0.7800          1.370260
        0.8400          1.301702
        0.9000          1.240806
        0.9600          1.186282
        1.0200          1.137124
        1.0800          1.092530
        1.1400          1.051856
        1.2000          1.014577
        1.2600          0.980258
        1.3200          0.948540
        1.3800          0.919119
        1.4400          0.891740
        1.5000          0.866183

Process returned 0 (0x0)   execution time : 16.117 s
Press any key to continue.
```

Q2)

```c
#include <stdio.h>
#include <string.h>
#define NUM_ROWS 20
#define TRUE 1
#define FALSE 0

typedef struct
{
    char name[100];
    double total;
    double max;
    double min;
    double r;
}FOREST;

void getForestInput(FOREST *);
void calculateTableData (FOREST *, double *, int *);
void displayTable (FOREST, double *, int *);

int main()
{
    FOREST fo;
    double uncut[20]; // uncut acres to start
    int years[20]; // number of years for total reforestation
    int flag;
    char R;

    do
    {
        flag = FALSE;
        getForestInput(&fo);
        calculateTableData (&fo, uncut, years);
        displayTable (fo, uncut, years);

        printf("Do you wish to quit (y/n)?\n");
        scanf("%s", &R);

```

```c
        if (R == 'n') flag = FALSE;

        while (R != 'n' && R != 'y')
        {
            printf("Do you wish to quit (y/n)?\n");
            scanf("%s", &R);
        }
        if (R == 'y')
        {
            printf("Program Terminated\n");
            break;
        }
    }while (flag == FALSE);
}

void getForestInput(FOREST *fo)
{
    int i, flag;
    double array[4] = {fo->total=0.0,fo->max=0.0,fo->min=0.0,fo->r=0.0};
    char variables[4][30] = {"Total acres", "Maximum uncut acres", "Minimum uncut acres", "Reforestation rate"};
    fflush(stdin);
    printf("Forest name: ");
    fgets(fo->name, 100, stdin);
    fo->name[strcspn(fo->name, "\n")] = '\0';

    for(i = 0;i <= 3; i++)
    {
        do
        {
            flag = TRUE;
            printf("%s: ", variables + i);
            scanf("%lf",&array[i]);
            if(array[i] <= 0)
            {
                printf("value must be greater than 0.\n");
                flag = FALSE;
            }

            if (array[2]>array[1]|| array[1]>array[0])
            {
                printf("The maximum uncut acres must be less than the total acres of the forest,\nand the minimum uncut acres must be less than both.\n");
                flag = FALSE;
            }

            if (array[3] >= 1)
            {
                printf("The reforestation rate must be between 0 and 1.\n");
                flag = FALSE;
            }

        } while(flag == FALSE);
    }

    fo->total = array[0];
    fo->max = array[1];
    fo->min = array[2];
    fo->r = array[3];
}

void calculateTableData(FOREST *fo, double *uncut, int *years)
{
    int x, i = 0;
    double A=0.0, prevA = 0.0;
    double increment = ((fo->max)-(fo->min))/(NUM_ROWS-1);

    for (x=0; x<NUM_ROWS ; x++)
    {
        uncut[x]= (fo->min) + (increment*x);
        while(A<(fo->total))
        {
            if (i==0)
                {
                    A=uncut[x];
                    prevA = A;
                }

```

```
114                 else
115                 {
116                     A = prevA + (fo->r)*prevA;
117                     prevA = A;
118                 }
119             i=i+1;
120         }
121         A = 0.0;
122         years[x]=i-1;
123         i = 0;
124     }
125 }
126
127 void displayTable(FOREST fo, double *uncut, int *years)
128 {
129     int i;
130     printf("\nForest: %s\n\tUncut Area:\tmin = %.2lf,\tmax = %.2lf\n\tReforestation Rate: %.4lf\n", fo.name, fo.min, fo.max, fo.r);
131     printf("\tUncut Area Start\t\tNum Years until total reforestation\n");
132
133     for(i=0; i<NUM_ROWS ; i++)
134     {
135         if (uncut[i] < 1000.0)
136         {
137             printf("\t\t%.3lf\t\t\t\t%d\n", uncut[i], years[i]);
138         }
139         else
140         {
141             printf("\t\t%.3lf\t\t\t%d\n", uncut[i], years[i]);
142         }
143     }
144 }
```

Test Cases:

```
Forest name: Forest 1
Total acres: 15000
Maximum uncut acres: 5000
Minimum uncut acres: 300
Reforestation rate: 0.05

Forest: Forest 1
        Uncut Area:     min = 300.00,   max = 5000.00
        Reforestation Rate: 0.0500
        Uncut Area Start                Num Years until total reforestation
                300.000                         81
                547.368                         68
                794.737                         61
                1042.105                        55
                1289.474                        51
                1536.842                        47
                1784.211                        44
                2031.579                        41
                2278.947                        39
                2526.316                        37
                2773.684                        35
                3021.053                        33
                3268.421                        32
                3515.789                        30
                3763.158                        29
                4010.526                        28
                4257.895                        26
                4505.263                        25
                4752.632                        24
                5000.000                        23
Do you wish to quit (y/n)?
l
Do you wish to quit (y/n)?
3432
Do you wish to quit (y/n)?
m
Do you wish to quit (y/n)?
n
Forest name: Forest 2
Total acres: 22676
Maximum uncut acres: 10000
Minimum uncut acres: 1000
Reforestation rate: 0.10
```

```
Forest: Forest 2
        Uncut Area:     min = 1000.00,  max = 10000.00
        Reforestation Rate: 0.1000
        Uncut Area Start                Num Years until total reforestation
                1000.000                        33
                1473.684                        29
                1947.368                        26
                2421.053                        24
                2894.737                        22
                3368.421                        21
                3842.105                        19
                4315.789                        18
                4789.474                        17
                5263.158                        16
                5736.842                        15
                6210.526                        14
                6684.211                        13
                7157.895                        13
                7631.579                        12
                8105.263                        11
                8578.947                        11
                9052.632                        10
                9526.316                        10
                10000.000                       9
Do you wish to quit (y/n)?
n
Forest name: Forest 3
Total acres: 58732
Maximum uncut acres: 60000
The maximum uncut acres must be less than the total acres of the forest,
and the minimum uncut acres must be less than both.
Maximum uncut acres: 20000
Minimum uncut acres: 23000
The maximum uncut acres must be less than the total acres of the forest,
and the minimum uncut acres must be less than both.
Minimum uncut acres: 5000
Reforestation rate: 3
The reforestation rate must be between 0 and 1.
Reforestation rate: -2
value must be greater than 0.
Reforestation rate: 0.15
```

```
Forest: Forest 3
        Uncut Area:     min = 5000.00,  max = 20000.00
        Reforestation Rate: 0.1500
        Uncut Area Start                Num Years until total reforestation
                5000.000                        18
                5789.474                        17
                6578.947                        16
                7368.421                        15
                8157.895                        15
                8947.368                        14
                9736.842                        13
                10526.316                       13
                11315.789                       12
                12105.263                       12
                12894.737                       11
                13684.211                       11
                14473.684                       11
                15263.158                       10
                16052.632                       10
                16842.105                       9
                17631.579                       9
                18421.053                       9
                19210.526                       8
                20000.000                       8
Do you wish to quit (y/n)?
y
Program Terminated

Process returned 0 (0x0)   execution time : 159.261 s
Press any key to continue.
```

Q3) note the '_' in console, indicates a space

A) CONSOLE

```
_11111111
__111111
___1111
____11
```

B) CONSOLE

```
1 0 1 4 3 7
```

C) CONSOLE

```
A is 12
arrayLength is 3
array values are: -1 20 30
A is 2
arrayLength is 3
array values are: -1 20 30
```

Q4)

printArray(b) → pass by reference
c=areTwoValuesTheSame(a, b[1]) → pass by value
IncreaseVale(&a) → pass by reference
shuffleArray(b) → pass by reference
d=isValueEven(a) → pass by value
joe(p) → pass by reference

Q5)

```c
#include <stdio.h>
#define UNKNOWN -1
typedef struct rectangle
{
 int width;
 int length;
 int area;
};
void printRectangle(struct rectangle A)
{
 if (A.width==UNKNOWN)
    printf("width is unknown\n");
    else
        printf("width is %d\n", A.width);
        if (A.length==UNKNOWN)
            printf("length is unknown\n");
            else
                printf("length is %d\n", A.length);
                if (A.area==UNKNOWN)
                    printf("area is unknown\n");
                    else
                        printf("area is %d\n", A.area);
}

void getRectangleDimension(struct rectangle *myBox)
{
    printf("what is the width?\n");
    scanf("%d", &myBox->width);

    printf("what is the length?\n");
    scanf("%d", &myBox->length);

    (*myBox).area = (myBox->length)*(myBox->width);
}

int main()
{
    struct rectangle myBox;
    myBox.width=UNKNOWN;
    myBox.length=UNKNOWN;
    myBox.area=UNKNOWN;
    getRectangleDimension(&myBox);
    printRectangle(myBox);
 return 0;
}
```

```
what is the width?
4
what is the length?
5
width is 4
length is 5
area is 20

Process returned 0 (0x0)   execution time : 228.767 s
Press any key to continue.
```