

Project Complete

Task 1

Description:

Creating a simple calculator in C programming using the
If selection statement,
switch Multiple selection statement,
while iteration Statement,
Counter Control iteration,

This program takes an arithmetic operator +, -, *, / and two operands from the user. Then, it performs the calculation on the two operands depending upon the operator entered by the user.

The * operator entered by the user is stored in op. And, the two operands, 1.5 and 4.5 are stored in first and second respectively.

Since the operator * matches case '*', the control of the program jumps to

```
printf("%.1f * %.1f = %.1f", first, second, first * second);
```

This statement calculates the product and displays it on the screen.

To make our output look cleaner, we have simply limited the output to one decimal place using the code %.1f.

Finally, the break; statement ends the switch statement

Code:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char op;
7      int i = 0;
8      double first, second;
9      while (op != 'q')
10     {
11         printf("\n\t\tEnter an operator (+, -, *, /)(q/ Quit): ");
12         if (i > 0)
13         {
14             char ch = getchar();
15         }
16         i++;
17         scanf("%c", &op);
18         if (op == 'q')
19         {
20             printf("Calculator is Terminating\n");
21             return 0;
22         }
23         printf("\nTwo operands: ");
24         printf("\nEnter first operands: ");
25         scanf("%lf", &first);
26         printf("Enter second operands: ");
27         scanf("%lf", &second);
28
29         switch (op)
30         {
31             case '+':
32                 printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
33                 break;
34             case '-':
35                 printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
36                 break;
37             case '*':
38                 printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
39                 break;
40             case '/':
41                 printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
42                 break;
```

```

16     l++;
17     scanf("%c", &op);
18     if (op == 'q')
19     {
20         printf("Calculator is Terminating\n");
21         return 0;
22     }
23     printf("\nTwo operands: ");
24     printf("\nEnter first operands: ");
25     scanf("%lf", &first);
26     printf("Enter second operands: ");
27     scanf("%lf", &second);
28
29     switch (op)
30     {
31     case '+':
32         printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
33         break;
34     case '-':
35         printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
36         break;
37     case '*':
38         printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
39         break;
40     case '/':
41         printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
42         break;
43         // operator doesn't match any case constant
44     default:
45         printf("Error! operator is not correct");
46     }
47 }
48 return 0;
49 }

```

Output:

```

D:\Fivern\Atif\C Project\Task1\calculator.exe
Enter an operator (+, -, *, /)(q/ Quit): +
Two operands:
Enter first operands: 2
Enter second operands: 3
2.0 + 3.0 = 5.0
Enter an operator (+, -, *, /)(q/ Quit): -
Two operands:
Enter first operands: 3
Enter second operands: 1
3.0 - 1.0 = 2.0
Enter an operator (+, -, *, /)(q/ Quit): *
Two operands:
Enter first operands: 3
Enter second operands: 1
3.0 * 1.0 = 3.0
Enter an operator (+, -, *, /)(q/ Quit): /
Two operands:
Enter first operands: 6
Enter second operands: 2
6.0 / 2.0 = 3.0
Enter an operator (+, -, *, /)(q/ Quit): q
Two operands:
Enter first operands: 2
Enter second operands: 4

```

```
Enter first operands: 0
Enter second operands: 2
6.0 / 2.0 = 3.0
Enter an operator (+, -, *, /)(q/ Quit): 0

Two operands:
Enter first operands: 2
Enter second operands: 4
Error! operator is not correct
Enter an operator (+, -, *, /)(q/ Quit): q
Calculator is Terminating

-----
Process exited after 67.04 seconds with return value 0
Press any key to continue . . .
```

omp

port Compilation

- Errors: 0
- Warnings: 0

Task 2

Number Converter:

Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

If else... selection statement,

Nested loop Counter statement

switch Multiple selection statement,

while iteration Statement,

Counter Control iteration,

For iteration statement,

Here is a program for numbers conversion. This program can convert Binary number, decimal number and hexadecimal number to each other respectively.

By entering choice, you can get choice for number conversion.

If user can choice 1, it will convert binary to decimal

For choice 1, it will take binary number and it will output will be in decimal number.

If user can choice 2, it will convert binary to hexa-decimal

For choice 2, it will take binary number and it will output will be in hexa-decimal number.

If user can choice 3, it will convert decimal to binary

For choice 3, it will take decimal number and it will output will be in binary number.

If user can choice 4, it will convert decimal to hexa-decimal

For choice 4, it will take decimal number and it will output will be in hexa-decimal number.

If user can choice 5, it will convert hexa-decimal to binary

For choice 5, it will take hexa-decimal number and it will output will be in binary number.

If user can choice 6, it will convert hexa-decimal to decimal

For choice 6, it will take hexa-decimal number and it will output will be in decimal number.

Code:

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <string.h>
4  #include <conio.h>
5
6  long int Binary_to_Decimal(long int);
7  long int Binary_to_Hexadecimal(long int);
8  long int Decimal_to_Binary(long int);
9  long int Decimal_to_Hexadecimal(long int);
10 void Hexadecimal_to_Binary(char[]);
11 void Hexadecimal_to_Decimal(char[]);
12
13 int main()
14 {
15     int operations, numbers = 1, checking;
16     long int b, o, d;
17     char h[100];
18     int x, y, spacebar;
19
20     printf("\t\tWELCOME TO NUMBER SYSTEM CONVERSION\n\n");
21
22     while (numbers != 0)
23     {
24         printf("\t\t>>>>>>> CHOOSE THE CONVERSION <<<<<\n\n");
25
26         printf("\n=> BINARY <=\n");
27         printf("1: Binary to Decimal.\n2: Binary to Hexa-Decimal.\n");
28
29         printf("\n=> DECIMAL <=\n");
30         printf("3: Decimal to Binary.\n4: Decimal to Hexa-Decimal.\n");
31
32         printf("\n=> HEXA-DECIMAL <=\n");
33         printf("5: Hexa-Decimal to Binary.\n6: Hexa-Decimal to Decimal.\n");
34
35         printf("\nENTER YOUR CHOICE: ");
36         scanf("%d", &operations);
37
38         switch (operations)
39         {
40             case 1:
41                 printf("\n***BINARY TO DECIMAL***\n");
42                 D:
43                 printf("\nEnter the Number in Binary form (0s & 1s): ");
44                 scanf("%ld", &b);
45
46                 checking = b;
47
48                 while (checking != 0)
49                 {
50                     numbers = checking % 10;
51                     if (numbers > 1)
52                     {
53                         printf("\n%d IS NOT BINARY NUMBER.\n", b);
54                         printf("***TRY AGAIN***\n");
55                         goto D;
56                     }
57                     else
58                         checking = checking / 10;
59                 }
60
61                 Binary_to_Decimal(b);
62                 break;
63
64             case 2:
65                 printf("\n***BINARY TO HEXA-DECIMAL***\n");
66                 F:
67                 printf("\nEnter the Number in Binary form (0s & 1s): ");
68                 scanf("%ld", &b);
69
70                 checking = b;
71
72                 while (checking != 0)

```

```

cimal() 73
xadeci 74
inary() 75
lexade 76
_to_Bin 77
_to_De 78
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

{
    numbers = checking % 10;
    if (numbers > 1)
    {
        printf("\n%d IS NOT BINARY NUMBER.\n", b);
        printf("****TRY AGAIN****\n");
        goto F;
    }
    else
        checking = checking / 10;
}

Binary_to_Hexadecimal(b);
break;

case 3:
    printf("\n***DECIMAL TO BINARY***\n");
    printf("\nEnter the Number in Decimal form (0 to 9): ");
    scanf("%ld", &d);
    Decimal_to_Binary(d);
    break;

case 4:
    printf("\n***DECIMAL TO HEXA-DECIMAL***\n");
    printf("\nEnter the Number in Decimal form (0 to 9): ");
    scanf("%ld", &d);
    Decimal_to_Hexadecimal(d);

    break;

case 5:
    printf("\n***HEXA-DECIMAL TO BINARY***\n");
X:
    printf("\nEnter the Number in Hexa-Decimal form: ");
    scanf("%s", &h);
    //check
    for (x = strlen(h) - 1; x >= 0; x--)
    {
        if (h[x] > 'f' && h[x] <= 'z' || h[x] > 'F' && h[x] <= 'Z')
        {
            printf("\nYou have to Enter Hexa-Decimal Number.\n");
            printf("%c IS NOT Hexa-Decimal Number.\n", h[x]);
            goto X;
        }
    }
    Hexadecimal_to_Binary(h);
    break;

case 6:
    printf("\n***HEXA-DECIMAL TO DECIMAL***\n");
Y:
    printf("\nEnter the Number in Hexa-Decimal form: ");
    scanf("%s", &h);
    //check
    for (x = strlen(h) - 1; x >= 0; x--)
    {
        if (h[x] > 'f' && h[x] <= 'z' || h[x] > 'F' && h[x] <= 'Z')
        {
            printf("\nYou have to Enter Hexa-Decimal Number.\n");
            printf("%c IS NOT Hexa-Decimal Number.\n", h[x]);
            goto Y;
        }
    }
    Hexadecimal_to_Decimal(h);

    break;

default:
    printf("\n***INVALID NUMBER***\n");
    break;
}
printf("\n\nDO YOU WANT TO CONTINUE = (1/0) :\n");

```



```

145 scanf("%d", &numbers);
146 }
147
148 spacebar = 3 + 35;
149 for (x = 1; x <= 3; x++)
150 {
151     for (y = 1; y <= spacebar; y++)
152     {
153         printf(" ");
154     }
155     spacebar--;
156     for (y = 1; y <= 2 * x - 1; y++)
157     {
158         printf("*");
159     }
160     printf("\n");
161 }
162 spacebar = 37;
163 for (x = 1; x <= 3; x++)
164 {
165     for (y = 1; y <= spacebar; y++)
166     {
167         printf(" ");
168     }
169     spacebar++;
170     for (y = 1; y <= 2 * (3 - x) - 1; y++)
171     {
172         printf("*");
173     }
174
175     printf("\n");
176 }
177 printf("\t\t BROUGHT TO YOU BY ITSOURCECODE.COM \n\t\t Log On now For More C/C++ Free Projects");
178
179
180 long int Binary_to_Decimal(long int bin)
181 {
182     int remainder, summation = 0, x = 0;
183     while (bin != 0)
184     {
185         remainder = bin % 10;
186         bin = bin / 10;
187         summation = summation + remainder * pow(2, x);
188         x++;
189     }
190
191     printf("\nEquivalent Decimal Number : %d", summation);
192 }
193
194 long int Binary_to_Hexadecimal(long int bin)
195 {
196     int remainder, x = 0, summation = 0, remaining[100], length = 0;
197
198     while (bin != 0)
199     {
200         remainder = bin % 10;
201         bin = bin / 10;
202         summation = summation + remainder * pow(2, x);
203         x++;
204     }
205     x = 0;
206     while (summation != 0)
207     {
208         remaining[x] = summation % 16;
209         summation = summation / 16;
210         x++;
211         length++;
212     }
213     printf("\nEquivalent Hexa-Decimal Number : ");
214     for (x = length - 1; x >= 0; x--)
215     {
216         switch (remaining[x])
217         {

```

```

217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

```

_Decimal()
 _Hexadeci
 to_Binary()
 to_Hexade
 imal_to_Bin
 imal_to_De
 nt

```

{
    case 10:
        printf("A");
        break;

    case 11:
        printf("B");
        break;

    case 12:
        printf("C");
        break;

    case 13:
        printf("D");
        break;

    case 14:
        printf("E");
        break;

    case 15:
        printf("F");
        break;

    default:
        printf("%d", remaining[x]);
}

}

}

long int Decimal_to_Binary(long int dec)
{
    int remainder[50], x, length = 0;
    do
    {
        remainder[x] = dec % 2;
        dec = dec / 2;
        x++;
        length++;
    } while (dec != 0);
}

```

```

259     printf("\nEquivalent Binary Number : ");
260     for (x = length - 1; x >= 0; x--)
261     {
262         printf("%d", remainder[x]);
263     }
264 }
265
266 long int Decimal_to_Hexadecimal(long int dec)
267 {
268     int remainder[50], x, length = 0;
269     do
270     {
271         remainder[x] = dec % 16;
272         dec = dec / 16;
273         x++;
274         length++;
275     } while (dec != 0);
276
277     printf("\nEquivalent Hexa-Decimal Number : ");
278     for (x = length - 1; x >= 0; x--)
279     {
280         switch (remainder[x])
281         {
282             case 10:
283                 printf("A");
284                 break;
285
286             case 11:
287                 printf("B");
288                 break;
289
290             case 12:
291                 printf("C");
292                 break;
293
294             case 13:
295                 printf("D");
296                 break;
297
298             case 14:
299                 printf("E");
300                 break;

```

```
313 {
314
315     int x = 0;
316     printf("\nEquivalent Binary Number : ");
317     for (x = 0; x < strlen(hex); x++)
318     {
319         switch (hex[x])
320         {
321             case '0':
322                 printf("0000");
323                 break;
324             case '1':
325                 printf("0001");
326                 break;
327             case '2':
328                 printf("0010");
329                 break;
330             case '3':
331                 printf("0011");
332                 break;
333             case '4':
334                 printf("0100");
335                 break;
336             case '5':
337                 printf("0101");
338                 break;
339             case '6':
340                 printf("0110");
341                 break;
342             case '7':
343                 printf("0111");
344                 break;
345             case '8':
346                 printf("1000");
347                 break;
348             case '9':
349                 printf("1001");
350                 break;
351             case 'A':
352                 printf("1010");
353                 break;
354             case 'B':
```

```

355     case 'b':
356         printf("1011");
357         break;
358     case 'C':
359     case 'c':
360         printf("1100");
361         break;
362     case 'D':
363     case 'd':
364         printf("1101");
365         break;
366     case 'E':
367     case 'e':
368         printf("1110");
369         break;
370     case 'F':
371     case 'f':
372         printf("1111");
373         break;
374
375     default:
376         printf("\n Invalid hexa digit %c ", hex[x]);
377     }
378 }
379 }
380
381 void Hexadecimal_to_Decimal(char hex[])
382 {
383     int x, numbers = 0, powered = 0, decimal = 0;
384
385     for (x = strlen(hex) - 1; x >= 0; x--)
386     {
387         if (hex[x] == 'A' || hex[x] == 'a')
388         {
389             numbers = 10;
390         }
391         else if (hex[x] == 'B' || hex[x] == 'b')
392         {
393             numbers = 11;
394         }
395         else if (hex[x] == 'C' || hex[x] == 'c')
396         {

```

```

382 {
383     int x, numbers = 0, powered = 0, decimal = 0;
384
385     for (x = strlen(hex) - 1; x >= 0; x--)
386     {
387         if (hex[x] == 'A' || hex[x] == 'a')
388         {
389             numbers = 10;
390         }
391         else if (hex[x] == 'B' || hex[x] == 'b')
392         {
393             numbers = 11;
394         }
395         else if (hex[x] == 'C' || hex[x] == 'c')
396         {
397             numbers = 12;
398         }
399         else if (hex[x] == 'D' || hex[x] == 'd')
400         {
401             numbers = 13;
402         }
403         else if (hex[x] == 'E' || hex[x] == 'e')
404         {
405             numbers = 14;
406         }
407         else if (hex[x] == 'F' || hex[x] == 'f')
408         {
409             numbers = 15;
410         }
411         else
412             //(a[i]>=0 || a[i]<=9)
413         {
414             numbers = hex[x] - 48;
415         }
416
417         decimal = decimal + numbers * pow(16, powered);
418         powered++;
419     }
420     printf("\nEquivalent Decimal Number : %d", decimal);
421 }

```

Output:

```
WELCOME TO NUMBER SYSTEM CONVERSION

>>>>> CHOOSE THE CONVERSION <<<<<<

=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.
=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.
=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 1

***BINARY TO DECIMAL***

Enter the Number in Binary form (0s & 1s): 1010101

Equivalent Decimal Number : 85

DO YOU WANT TO CONTINUE = (1/0) :
1

>>>>> CHOOSE THE CONVERSION <<<<<<
```

Compilation: Success
- Errors: 0
Warnings: 0

```
=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.
=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 3

***DECIMAL TO BINARY***

Enter the Number in Decimal form (0 to 9): 101

Equivalent Binary Number : 1100101

DO YOU WANT TO CONTINUE = (1/0) :
4

>>>>> CHOOSE THE CONVERSION <<<<<<

=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.
=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.
```

Compilation: Success

```

Bi=> HEXA-DECIMAL <=
Bi5: Hexa-Decimal to Binary.
D6: Hexa-Decimal to Decimal.
D
HENTER YOUR CHOICE: 20
He
m***INVALID NUMBER***

DO YOU WANT TO CONTINUE = (1/0) :
1
>>>>> CHOOSE THE CONVERSION <<<<<

=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.

=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.
ComENTER YOUR CHOICE: 4

```

```

Abort Compilation
-----
- Errors: 0
- Warnings: 0
- Output File name: D:\Binary\3rd\3rd\Task\Task\main.exe

```

```

t***DECIMAL TO HEXA-DECIMAL***
BiEnter the Number in Decimal form (0 to 9): 19
Bi
Dequivalent Hexa-Decimal Number : 13
D
HDO YOU WANT TO CONTINUE = (1/0) :
H2
m>>>>> CHOOSE THE CONVERSION <<<<<

=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.

=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 2

***BINARY TO HEXA-DECIMAL***
Enter the Number in Binary form (0s & 1s): 10001
m
Dequivalent Hexa-Decimal Number : 11

```

```

Abort Compilation
-----
- Errors: 0
- Warnings: 0
- Output File name: D:\Binary\3rd\3rd\Task\Task\main.exe

```



```
DO YOU WANT TO CONTINUE = (1/0) :
1
>>>>> CHOOSE THE CONVERSION <<<<<
Bi=> BINARY <=
D1: Binary to Decimal.
D2: Binary to Hexa-Decimal.
He=> DECIMAL <=
H3: Decimal to Binary.
H4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 5

***HEXA-DECIMAL TO BINARY***

Enter the Number in Hexa-Decimal form: AB

Equivalent Binary Number : 10101011

DO YOU WANT TO CONTINUE = (1/0) :
0

*
***
*****
***
```

TASK 3

Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

switch Multiple selection statement,

while iteration Statement,

Counter Control iteration,

This program is simple temperature converter.

There is choice option for each conversion, if you want conversion Celsius to Fahrenheit the enter 1 and so on.

Enter 0 to Quit or ending the program

Enter 1. Fahrenheit To Celsius

Enter 2. Celsius To Fahrenheit

Enter 3. Celsius To Kelvin

Enter 4. Kelvin To Celsius

If user Enter 0, it will quit program.

If user Enter 1, it will ask for Fahrenheit temperature and print temperature in Celsius.

If user Enter 2, it will ask for Celsius temperature and print temperature in Fahrenheit

If user Enter 3, it will ask for Celsius temperature and print temperature in Kelvin.

If user Enter 4, it will ask for Kelvin temperature and print temperature in Celsius

Code:

```

1  #include <stdio.h>
2
3  int main()
4  {
5
6      int run = 0;
7      int choice = -1;
8
9      while (choice != 0)
10     {
11         printf("\nEnter 0. Quit\n");
12         printf("Enter 1. Farhenheit To Celsius\n");
13         printf("Enter 2. Celsuis To Frahenheit\n");
14         printf("Enter 3. Celsuis To Kelvin \n");
15         printf("Enter 4. Kelvin To Celsuis\n");
16         printf("Enter Choice: ");
17         scanf("%d", &choice);
18
19         switch (choice)
20         {
21             case 0:
22                 printf("Quiting ");
23                 return 0;
24                 break;
25             case 1:
26             {
27                 double f, c;
28                 printf("Enter fahrenheit: ");
29                 scanf("%lf", &f);
30
31                 c = (f - 32) * 0.5556;
32                 printf("Celsius: %.2lf\n", c);
33             }
34
35             break;
36             case 2:
37             {
38                 double ff, cc;
39                 printf("Enter celsius:");
40                 scanf("%lf", &cc);
41
42                 ff = cc * 1.8 + 32;

```

```

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
}

break;
case 2:
{
    double ff, cc;
    printf("Enter celsius:");
    scanf("%lf", &cc);

    ff = cc * 1.8 + 32;
    printf("Fahrenheit:%.2lf\n", ff);
}

break;
case 3:
{
    float celsuis;
    printf("Enter temperature in celsius: ");
    scanf("%f", &celsuis);
    float kelvin;
    kelvin = 273.15 + celsuis;
    printf("%.2f Celsius = %.2f Kelvin\n", celsuis, kelvin);
}
break;
case 4:
{
    float kelvin;
    printf("Enter temperature in kelvin: ");
    scanf("%f", &kelvin);
    float celsius;
    celsius = 273.15 - kelvin;
    printf("%.2f Kelvin = %.2f Celsius\n", kelvin, celsius);
}
break;

default:
    printf("Invalid Choice \n");
    break;
}

return 0;
}

```

Output:

```
Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 1
Enter fahrenheit: 23
Celsius: -5.00

Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 2
Enter celsius:37
Fahrenheit:98.60

Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 3
Enter temperature in celsius: 28
28.00 Celsius = 301.15 Kelvin

Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 0
Quitting
-----
Process exited after 34.31 seconds with return value 0
Press any key to continue . . .

- Compilation Time: 0.34s
```

```
Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 4
Enter temperature in kelvin: 300
300.00 Kelvin = -26.85 Celsius

Enter 0. Quit
Enter 1. Fahrenheit To Celsius
Enter 2. Celsius To Fahrenheit
Enter 3. Celsius To Kelvin
Enter 4. Kelvin To Celsius
Enter Choice: 0
Quitting
-----
Process exited after 34.31 seconds with return value 0
Press any key to continue . . .

- Compilation Time: 0.34s
```

Task4

Areas

Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

If else... selection statement,

Counter Control iteration,

For iteration statement,

This program takes an arithmetic choice 1, 2, 3, 4 and 5

By choice of 1, it will ask for radius of a circle and find area of that circle.

By choice of 2, it will ask for radius of circle and find out perimeter of circle.

By choice of 3, it will ask for length and width of rectangle and find out area of rectangle.

By choice of 4, it will ask for length and width of rectangle and find out perimeter of rectangle.

At the end it has choice of 5 to quit program.

Code:

```

#include <stdio.h>
#include <stdbool.h>

float areaOfCircle(int rad)
{
    return (3.14 * rad * rad);
}

float periOfCircle(int rad)
{
    return (2 * 3.14 * rad);
}

int areaOfReactangle(int len, int wid)
{
    return len * wid;
}

int periOfRectangle(int len, int wid)
{
    return (2 * (len + wid));
}

int main()
{
    /* 2D array declaration*/
    while (true)
    {
        printf("\n\nEnter 1. For Area of Circle \n");
        printf("Enter 2. For Perimeter of Circle \n");
        printf("Enter 3. For Area of Rectangle \n");
        printf("Enter 4. For Perimeter of Rectangle \n");
        printf("Enter 5. To Quit \n");
        printf("Enter : ");
        int choice;
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:

```

```
printf("\n\nEnter 1. For Area of Circle \n");
printf("Enter 2. For Perimeter of Circle \n");
printf("Enter 3. For Area of Rectangle \n");
printf("Enter 4. For Perimeter of Rectangle \n");
printf("Enter 5. To Quit \n");
printf("Enter : ");
int choice;
scanf("%d", &choice);
switch (choice)
{
case 1:
{
    int rad;
    printf("Enter Raduis : ");
    scanf("%d", &rad);
    printf("Area of Circle is : %.2f", areaOfCircle(rad));
}

break;
case 2:
{
    int r;
    printf("Enter Raduis : ");
    scanf("%d", &r);
    printf("Perimeter of Circle is : %.2f", periOfCircle(r));
}
break;
```



```

56     break;
57
58     case 3:
59     {
60         int l, w;
61         printf("Enter length : ");
62         scanf("%d", &l);
63         printf("Enter width : ");
64         scanf("%d", &w);
65         printf("Area of Rectangle : %d", areaOfReactangle(l, w));
66     }
67     break;
68     case 4:
69     {
70
71         int len, wid;
72         printf("Enter length : ");
73         scanf("%d", &len);
74         printf("Enter width : ");
75         scanf("%d", &wid);
76         printf("Area of Rectangle : %d", periOfRectangle(len, wid));
77     }
78     break;
79     case 5:
80         return 0;
81         break;
82     // operator doesn't match any case constant
83     default:
84         printf("Error! Choice is not correct");
85     }
86 }
87
88 return 0;
89 }

```

OUTPUT:

```

Enter 1. For Area of Circle
Enter 2. For Perimeter of Circle
include <stdio.h> Enter 3. For Area of Rectangle
include <math.h> Enter 4. For Perimeter of Rectangle
Enter 5. To Quit
int areaOfCircle()
{
    Enter : 1
    Enter Raduis : 23
    return Area of Circle is : 1661.060059
}

int perimeterOfCircle()
{
    Enter 1. For Area of Circle
    Enter 2. For Perimeter of Circle
    return Enter 3. For Area of Rectangle
    Enter 4. For Perimeter of Rectangle
    Enter 5. To Quit
}

int areaOfRectangle()
{
    Enter : 4
    Enter length : 5
    return Enter width : 6
    Area of Rectangle : 22
}

int perimeterOfRectangle()
{
    Enter 1. For Area of Circle
    Enter 2. For Perimeter of Circle
    Enter 3. For Area of Rectangle
    Enter 4. For Perimeter of Rectangle
    Enter 5. To Quit
}

int main()
{
    Enter : 2
    /* 2D d
    while (1)
    {
        Enter Raduis : 2
        Perimeter of Circle is : 12.560000
        printf("Enter 2. For Perimeter of Circle \n");
        printf("Enter 3. For Area of Rectangle \n");
        printf("Enter 4. For Perimeter of Rectangle \n");
        printf("Enter 5. To Quit \n");
    }
}

```

```

Perimeter of Circle is : 12.560000
Enter 1. For Area of Circle
Enter 2. For Perimeter of Circle
include <stdio.h> Enter 3. For Area of Rectangle
include <math.h> Enter 4. For Perimeter of Rectangle
Enter 5. To Quit
int areaOfCircle()
{
    Enter : 4
    Enter length : 5
    return Enter width : 6
    Area of Rectangle : 22
}

int perimeterOfCircle()
{
    Enter 1. For Area of Circle
    return Enter 2. For Perimeter of Circle
    Enter 3. For Area of Rectangle
    Enter 4. For Perimeter of Rectangle
    Enter 5. To Quit
}

int areaOfRectangle()
{
    Enter : 5
}

int perimeterOfRectangle()
{
}

Process exited after 38.54 seconds with return value 0
Press any key to continue . . .

int main()
{
    /* 2D d
    while (1)
    {
        printf("Enter 2. For Perimeter of Circle \n");
        printf("Enter 3. For Area of Rectangle \n");
        printf("Enter 4. For Perimeter of Rectangle \n");
        printf("Enter 5. To Quit \n");
        printf("Enter : ");
        int choice;
        scanf("%d", &choice);
        switch (choice)
        {

```