

# Programming for Engineers (872H1)

## Workshop2 - Solutions

### 1) Create a user menu using if-else

```
#include <stdio.h>

#include <math.h>

int main()
{
    // declare and initialise variables and constants
    int menu_choice = -1;
    float input;
    float result;
    while (menu_choice != 0) {
        // print menu
        printf("Welcome. Make a choice:\n\n");
        printf("1 - Circumference of a circle\n");
        printf("2 - Square root of a number\n");
        printf("3 - Area of a square\n");
        printf("4 - Tan inverse of an input number\n");
        printf("0 - Exit\n");
        // elicit user input
        scanf("%d", &menu_choice);
        if (menu_choice==1)
        {
            printf("Enter radius of circle: ");
            scanf("%f",&input); //User input for radius
            result = 2 * 3.14159 * input; //Computing result
            printf("The circumference of circle is %0.2f\n",result);
        }
        else if (menu_choice==2)
        {
            printf("Enter a number: ");
            scanf("%f",&input);
```

## Programming for Engineers (872H1)

```
        result = sqrt(input); //sqrt function is present in math.h
library
        printf("The square root is %0.2f\n",result);
    }
    else if (menu_choice==3)
    {
        printf("Enter the dimension of square: ");
        scanf("%f",&input);
        result =input * input;
        printf("The area of square is %0.2f\n",result);
    }
    else if (menu_choice==4)
    {
        printf("Enter an input number: ");
        scanf("%f",&input);
        result =atan(input);

        /*atan function computes tan inverse and results in corresponding
        angle in radians. The function is present in math.h */

        printf("Tan inverse of the number is %0.2f\n",result);
    }
} //Ends while loop
return 0;
} //Ends main
```

## Programming for Engineers (872H1)

### 2) Create a user menu using switch

```
#include <stdio.h>

#include <math.h>

int main()

{

// declare and initialise variables and constants

char menu_choice = 'm';

float input;

float result;

while (menu_choice != 'e') {

    // print menu

    printf("Welcome. Make a choice:\n\n");

    printf("a - Circumference of a circle\n");

    printf("b - Square root of a number\n");

    printf("c - Area of a square\n");

    printf("d - Tan inverse of an input number\n");

    printf("e - Exit\n");

    // elicit user input

    scanf("%c", &menu_choice);

    switch(menu_choice){

    case 'a':

    {

        printf("Enter radius of circle: ");

        scanf("%f",&input); //User input for radius

        result = 2 * 3.14159 * input; //Computing result

        printf("The circumference of circle is %0.2f\n",result);

        getchar();

        break;

    }

    case 'b':

    {

        printf("Enter a number: ");

        scanf("%f",&input);
```

## Programming for Engineers (872H1)

```
    result = sqrt(input);
    printf("The square root is %0.2f\n",result);
    getchar();
    break;
}
case 'c':
{
    printf("Enter the dimension of square: ");
    scanf("%f",&input);
    result =input * input;
    printf("The area of square is %0.2f\n",result);
    getchar();
    break;
}
case 'd':
{
    printf("Enter an input number: ");
    scanf("%f",&input);
    result =atan(input);
    printf("Tan inverse of the number is %0.2f\n",result);
    getchar();
    break;
}
} //Ends switch
} //Ends while loop
return 0;
} //Ends main
```

## Programming for Engineers (872H1)

### 3) Recognising and avoiding mistakes – logic errors

- a. Logic errors indicated in RED below. The correct result is **889**.
- i. wrong order of computation - we need brackets
  - ii. wrong operator (previous result was overwritten)
  - iii. wrong output formatting symbol (character instead of decimal)

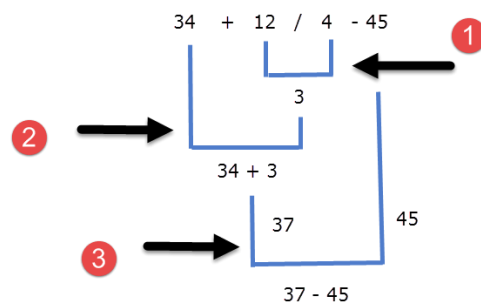
```
#include <stdio.h>

int main() {
    // declare and initialise variables and constants
    const int a = 5;
    const int b = 18;
    int result = 0;
    // compute (a+b)^2
    result = (a + b) * (a + b);
    // add 4ab and update result
    result += 4 * a * b;
    printf("(%d + %d)^2 + 4*%d*%d is %d", a,b,a,b,result);
    getchar();
    return 0;
}
```

## Programming for Engineers (872H1)

b.

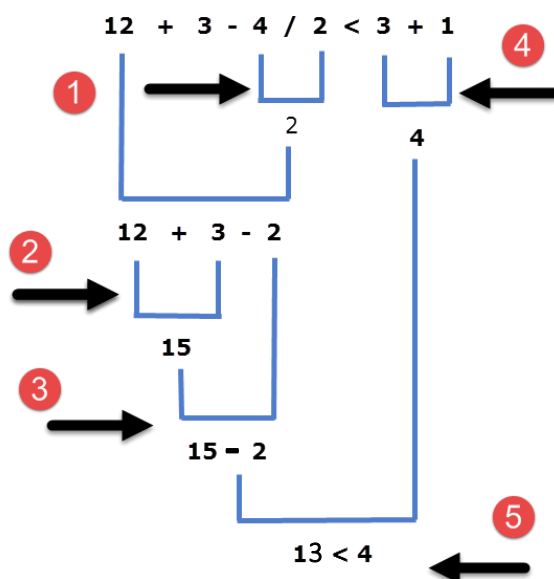
i.  $34 + 12 / 4 - 45$



Ans : -8

Here  $/$  has higher precedence hence  $12/4$  is evaluated first. The operators  $+$  and  $-$  have the same precedence, so which one of them will be evaluated first? To solve this problem you need to consult the associativity of the operator. The operators  $+$  and  $-$  have the same precedence and associates from left to right therefore in our expression  $34 + 12/4 - 45$  after division, addition( $+$ ) will be performed before subtraction( $-$ ) .

ii.  $12 + 3 - 4 / 2 < 3 + 1$



Ans: 0

Here  $/$  has higher precedence hence  $4/2$  is evaluated first. Then  $+$  and  $-$  operators have same precedence and associates from left to right, therefore in the expression  $12 + 3 - 4 / 2 < 3 + 1$  after division,  $+$  will be evaluated before  $-$ . The precedence of  $<$  operator is lower than that of  $/$ ,  $+$  and  $-$  hence it will be evaluated at last. As  $13 < 4$  is false, it will result in logic 0.

## Programming for Engineers (872H1)

### 4) Decimal to binary conversion

```
#include <stdio.h>

int main() {
    int dec; //input variable
    int binary[100]; //arbitrary array size for output
    int i=0; //Variable used for indexing
    printf("Enter a positive decimal number: ");
    scanf("%d",&dec);
    while (dec>1)
    {
        binary[i]= dec%2; //Remainder will be either 0 or 1
        dec=dec/2;
        i++;
    }
    binary[i]=1;
    printf("The binary equivalent is ");
    while(i>=0)
    {
        printf("%d",binary[i]); //Last value of array is most
significant bit
        i--;
    }
}
```