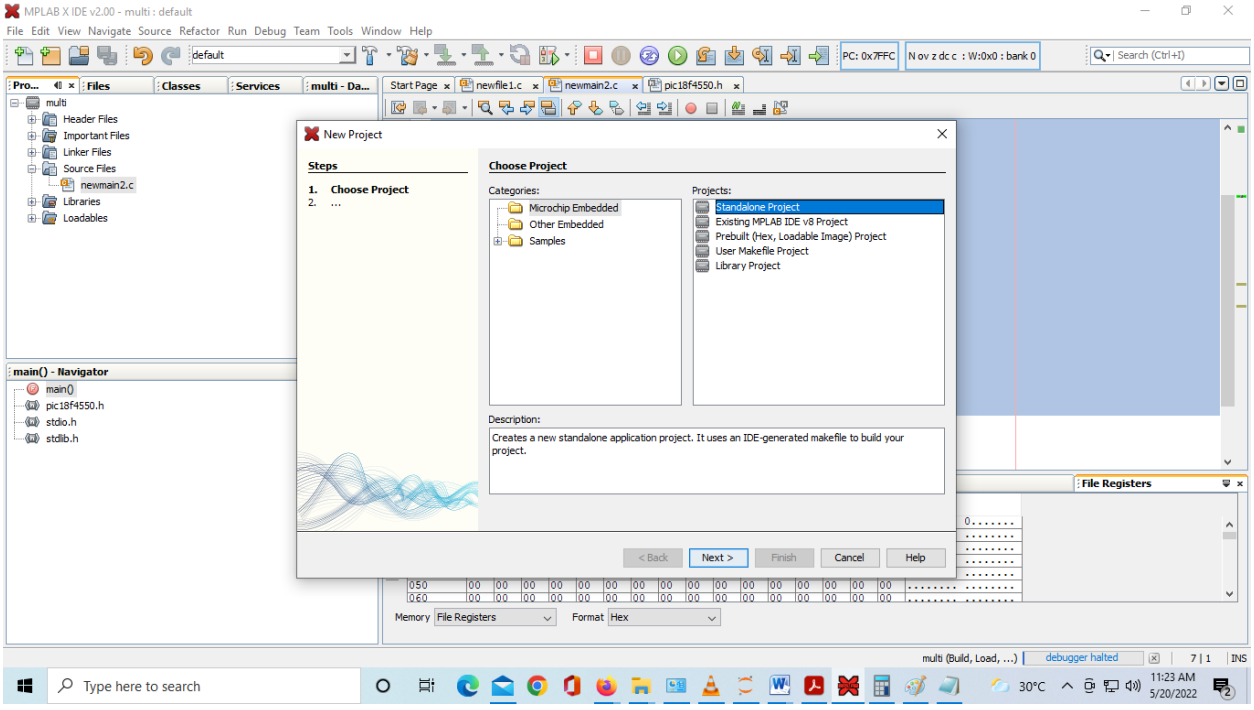
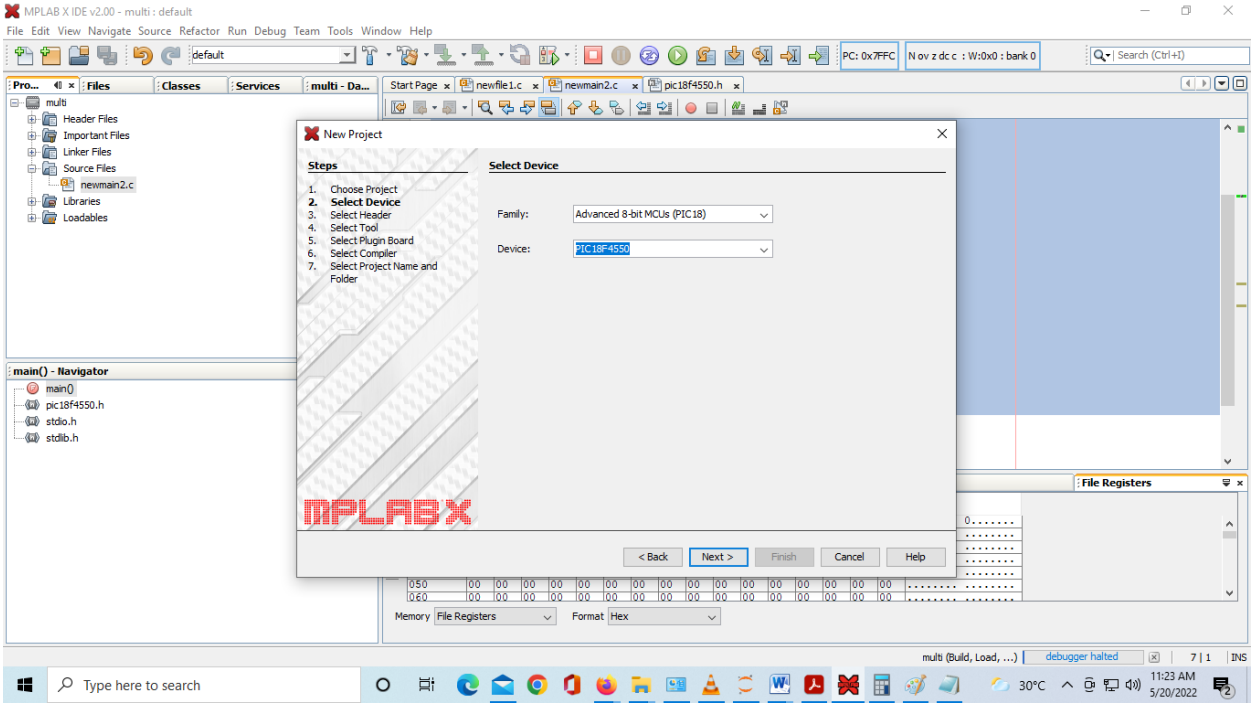
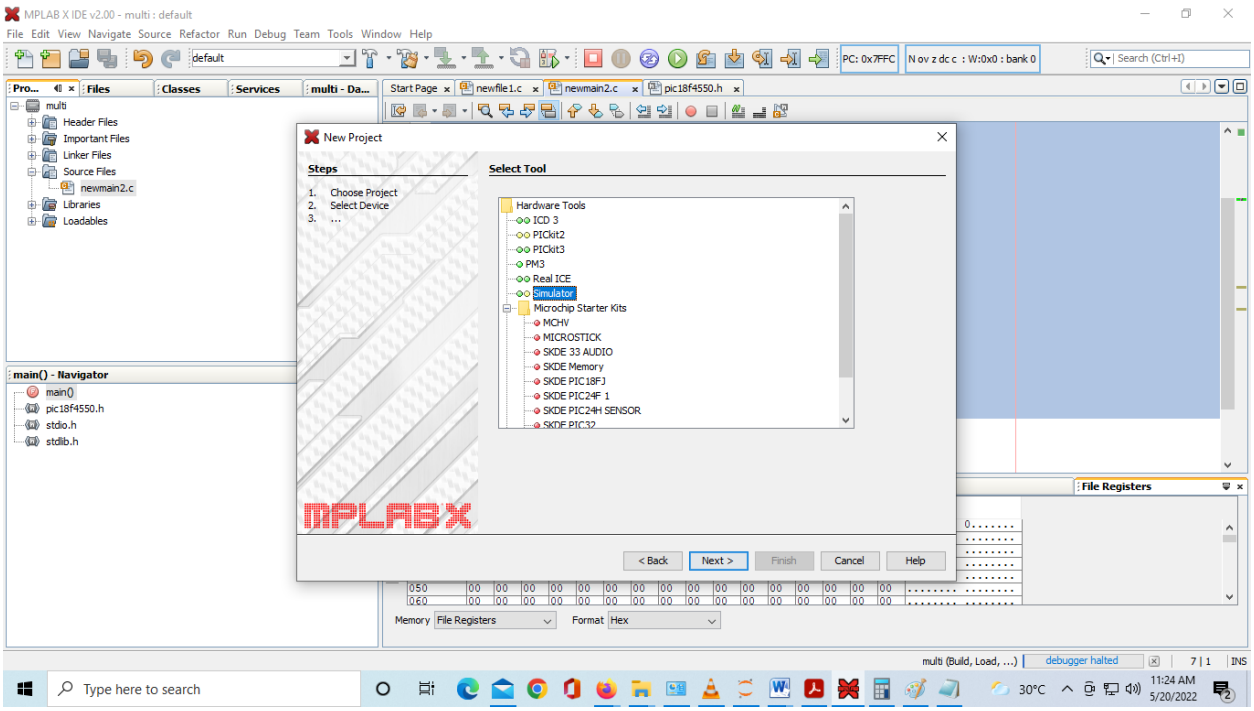
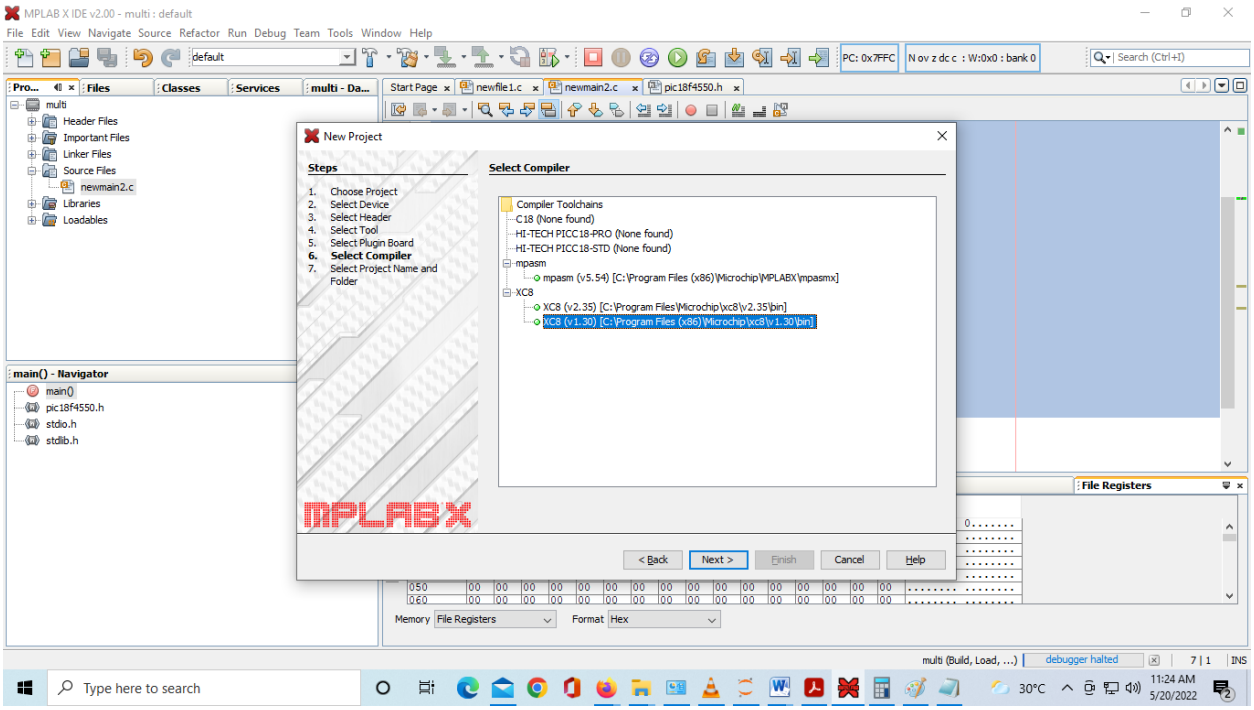
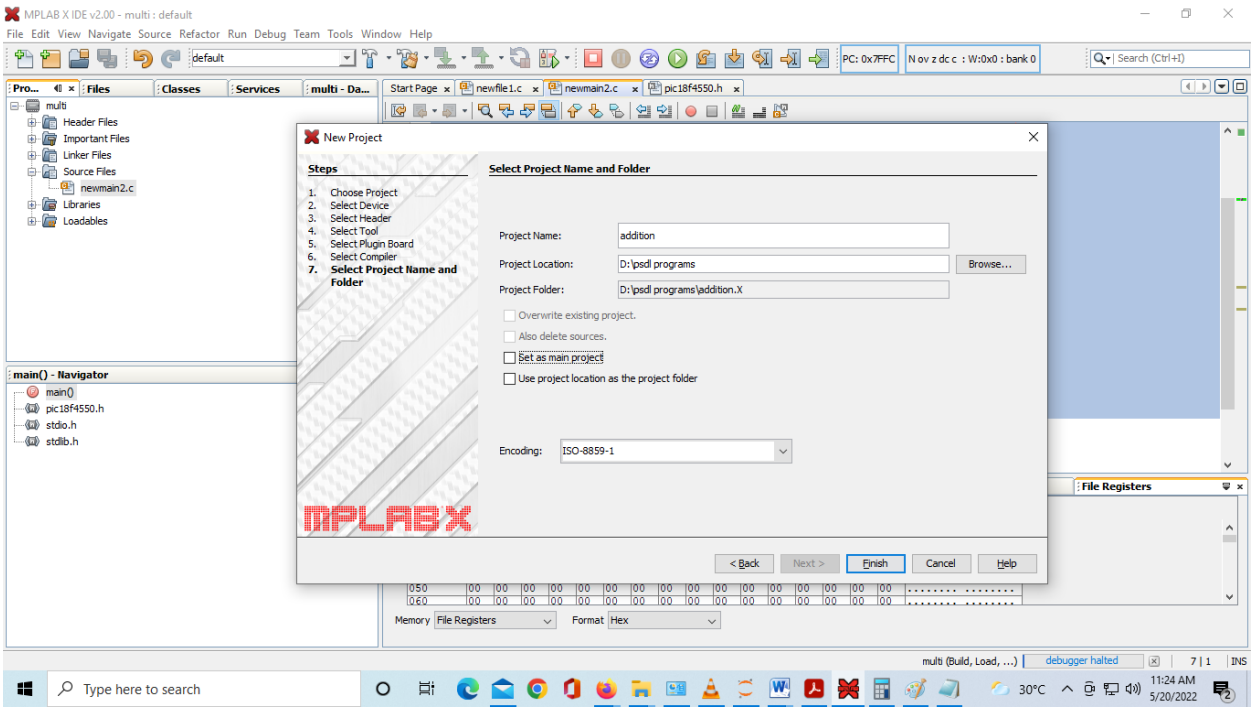
# Assignment 1:- Addition of numbers

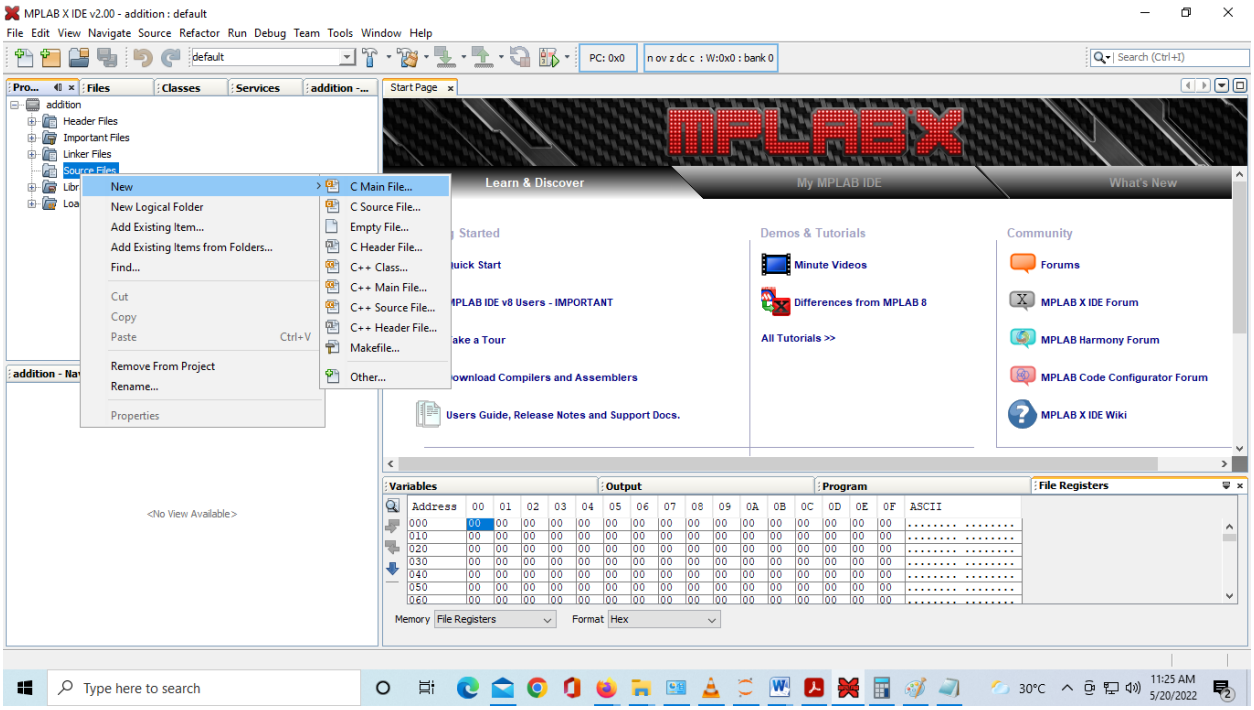


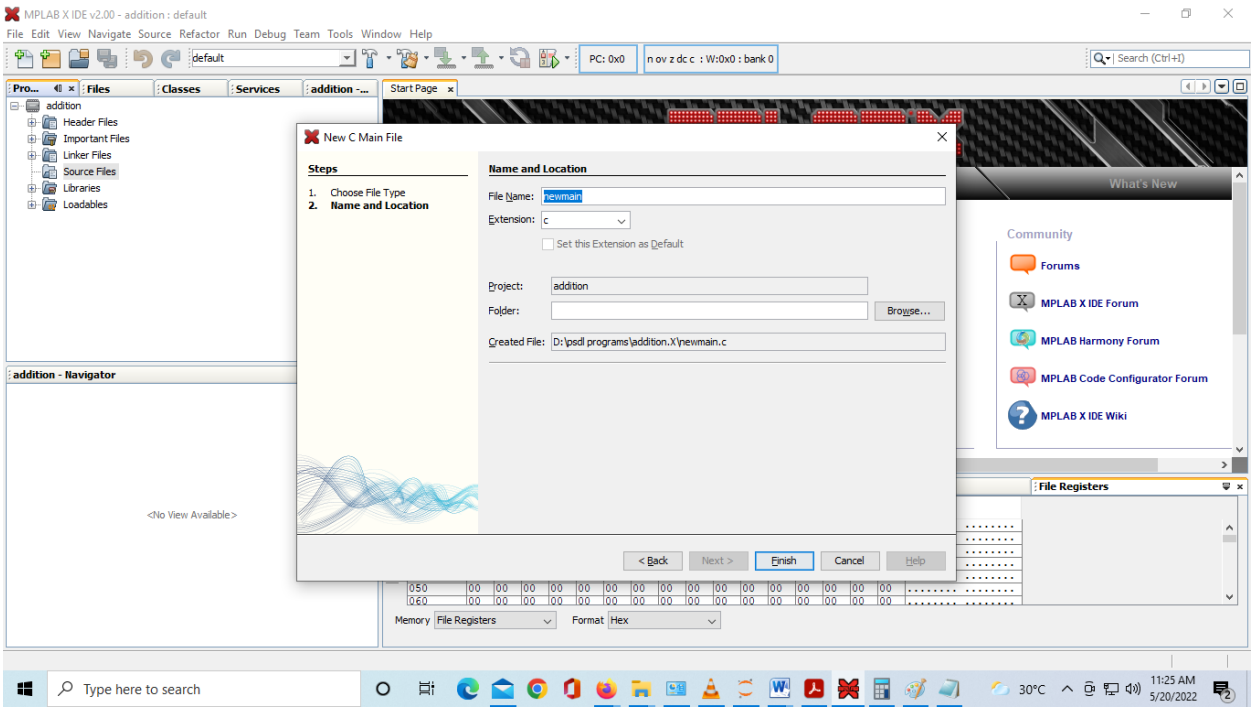




MPLAB X IDE v2.00 - multi : default
New Project
Steps
1. Choose Project
2. Select Device
3. ...
Select Tool
Hardware Tools
ICD 3
PICkit2
PICkit3
PM3
Real ICE
Simulator
Microchip Starter Kits
MCHV
MICROSTICK
SKDE 33 AUDIO
SKDE Memory
SKDE PIC18FJ
SKDE PIC24F 1
SKDE PIC24H SENSOR
SKDE PIC32
< Back
Next >
Finish
Cancel
Help


MPLAB X IDE v2.00 - multi : default
New Project
Steps
1. Choose Project
2. Select Device
3. Select Header
4. Select Tool
5. Select Plugin Board
6. Select Compiler
7. Select Project Name and Folder
Select Compiler
Compiler Toolchains
C18 (None found)
HI-TECH PICC18-PRO (None found)
HI-TECH PICC18-STD (None found)
mpasm
mpasm (v5.54) [C:\Program Files (x86)\Microchip\MPLABX\mpasmx]
XC8
XC8 (v2.35) [C:\Program Files\Microchip\xc8\v2.35\bin]
XC8 (v1.30) [C:\Program Files (x86)\Microchip\xc8\v1.30\bin]
< Back
Next >
Finish
Cancel
Help

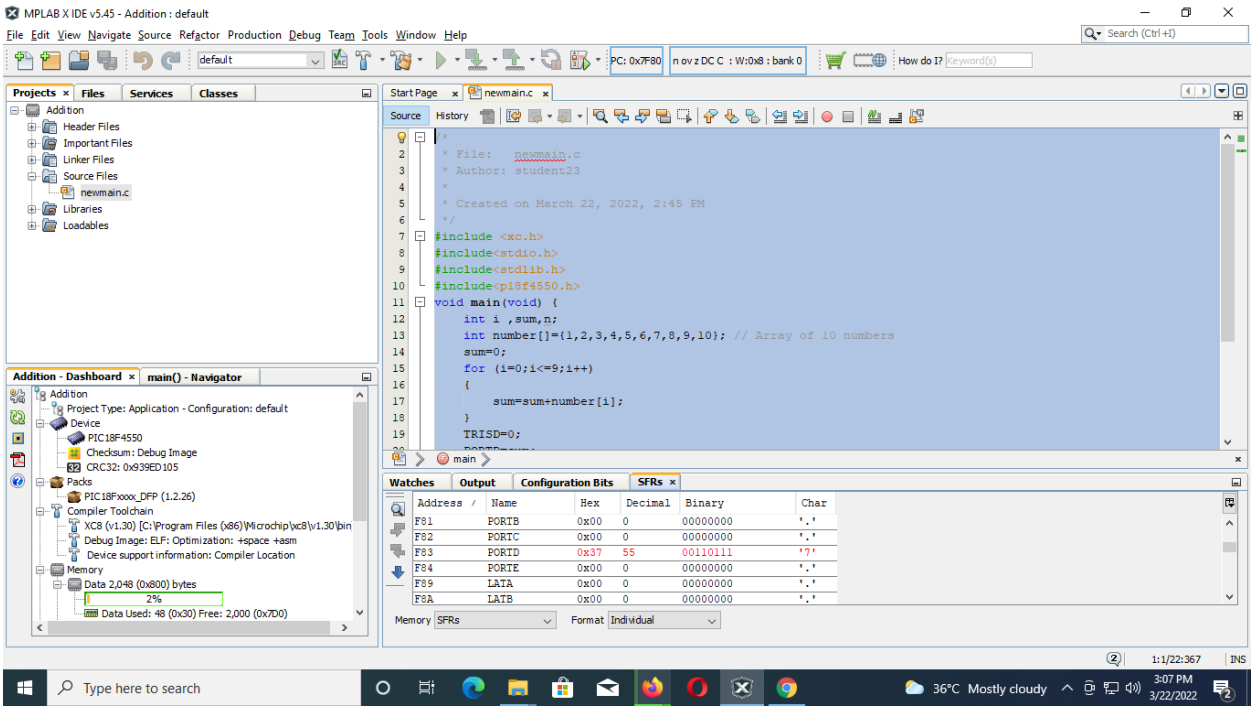

MPLAB X IDE v2.00 - multi : default
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
New Project
Steps
1. Choose Project
2. Select Device
3. Select Header
4. Select Tool
5. Select Plugin Board
6. Select Compiler
7. Select Project Name and Folder
Select Project Name and Folder
Project Name: addition
Project Location: D:\psdl programs
Project Folder: D:\psdl programs\addition.X
Browse...
Overwrite existing project.
Also delete sources.
Set as main project
Use project location as the project folder
Encoding: ISO-8859-1
< Back
Next >
Finish
Cancel
Help
main() - Navigator
File Registers
Memory File Registers
Format Hex
Type here to search
11:24 AM
5/20/2022


MPLAB X IDE v2.00 - addition : default
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
addition
Header Files
Important Files
Linker Files
Source Files
New
New Logical Folder
Add Existing Item...
Add Existing Items from Folders...
Find...
Cut
Copy
Paste
Remove From Project
Rename...
Properties
C Main File...
C Source File...
Empty File...
C Header File...
C++ Class...
C++ Main File...
C++ Source File...
C++ Header File...
Makefile...
Other...
<No View Available>
Start Page
Learn & Discover
My MPLAB IDE
What's New
Demos & Tutorials
Minute Videos
Differences from MPLAB 8
All Tutorials >>
Community
Forums
MPLAB X IDE Forum
MPLAB Harmony Forum
MPLAB Code Configurator Forum
MPLAB X IDE Wiki
Users Guide, Release Notes and Support Docs.
Variables
Output
Program
File Registers
Memory File Registers
Format Hex
Type here to search
11:25 AM
5/20/2022




```
/*
 * File:   newmain.c
 * Author: student23
 *
 * Created on March 22, 2022, 2:45 PM
 */
#include <xc.h>
#include<stdio.h>
#include<stdlib.h>
#include<p18f4550.h>
void main(void) {
    int i ,sum,n;
    int number[]={1,2,3,4,5,6,7,8,9,10}; // Array of 10 numbers
    sum=0;
    for (i=0;i<=9;i++)
    {
        sum=sum+number[i];
```

```
    }
    TRISD=0;
    PORTD=sum;
}
```




Our Output is 37 in Hexadecimal

```
1  #include <xc.h>
2  #include<stdio.h>
3  #include<stdlib.h>
4  #include<pl8f4550.h>
5  void main(void) {
6      int i ,sum,n;
7      int number[]={1,2,3,4,5,6,7,8,9,10}; // Array of 10 numbers
8      sum=0;
9      for (i=0;i<=9;i++)
10     {
11         sum=sum+number[i];
12     }
13     TRISD=0;
14     PORTD=sum;
15 }
16
```

Our Output is 37

Variables | Call Stack | Output | Program | File R

| Address | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | 00 | 12 | 00 | 0A | 00 | 01 | 00 | 02 | 00 | 03 | 00 | 04 | 00 | 05 | 00 | 06 | ..... ..... |
| 010 | 00 | 07 | 00 | 08 | 00 | 09 | 00 | 0A | 00 | 37 | 00 | 0A | 00 | 01 | 00 | 02 | ..... .7..... |
| 020 | 00 | 03 | 00 | 04 | 00 | 05 | 00 | 06 | 00 | 07 | 00 | 08 | 00 | 09 | 00 | 0A | ..... ..... |
| 030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... ..... |
| 040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... ..... |

Output can be seen in SFR register

Variables | Call Stack | Output | Program | SFR

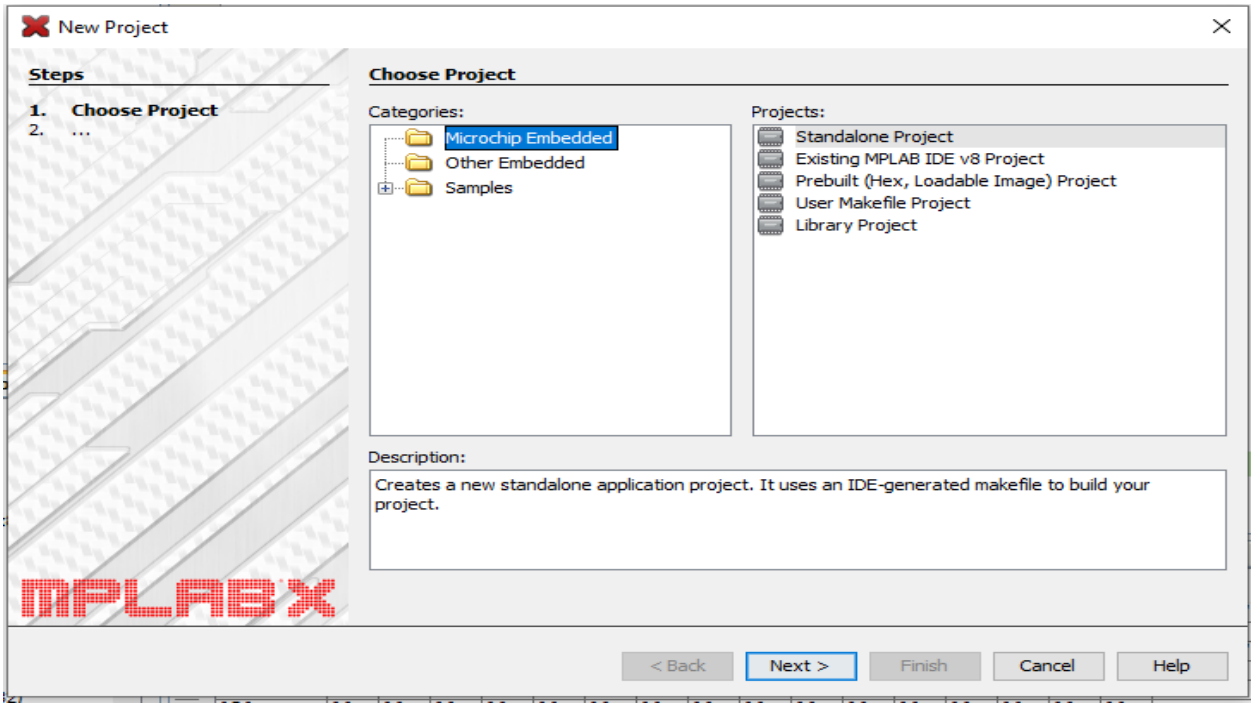
| Address | Name | Hex | Decimal | Binary | Char |
|---|---|---|---|---|---|
| F81 | PORTB | 0x00 | 0 | 00000000 | '.' |
| F82 | PORTC | 0x00 | 0 | 00000000 | '.' |
| F83 | PORTD | 0x37 | 55 | 00110111 | '7' |
| F84 | PORTE | 0x00 | 0 | 00000000 | '.' |
| F89 | LATA | 0x00 | 0 | 00000000 | '.' |
| F8A | LATB | 0x00 | 0 | 00000000 | '.' |
| F8B | LATC | 0x00 | 0 | 00000000 | '.' |

Output can be seen in SFR

Memory SFR  Format Individual

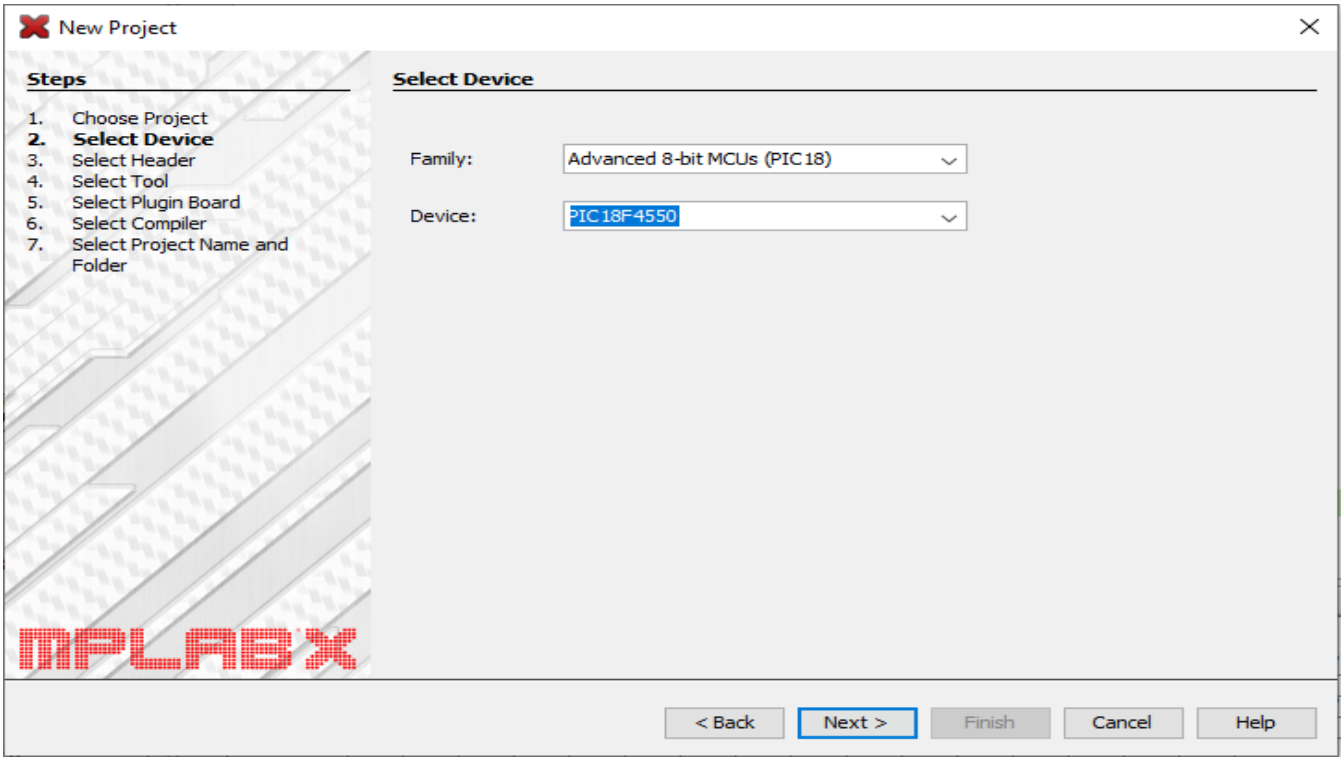
**Assignment 2:- Memory Transfer (Internal to Internal)**

**Step 1 :-** File → New Project → Select Microchip Embedded → Standalone Project

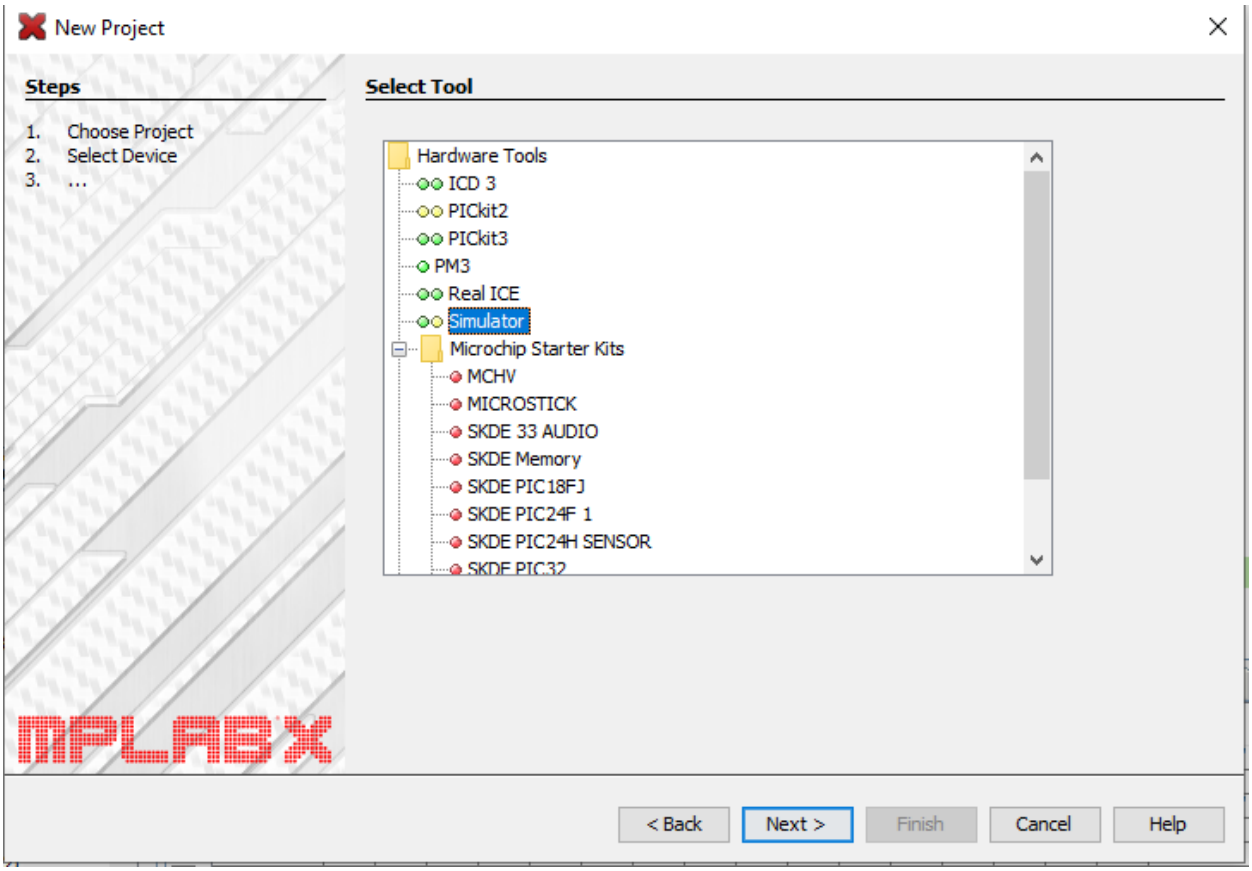


**Step 2:- Select Device**

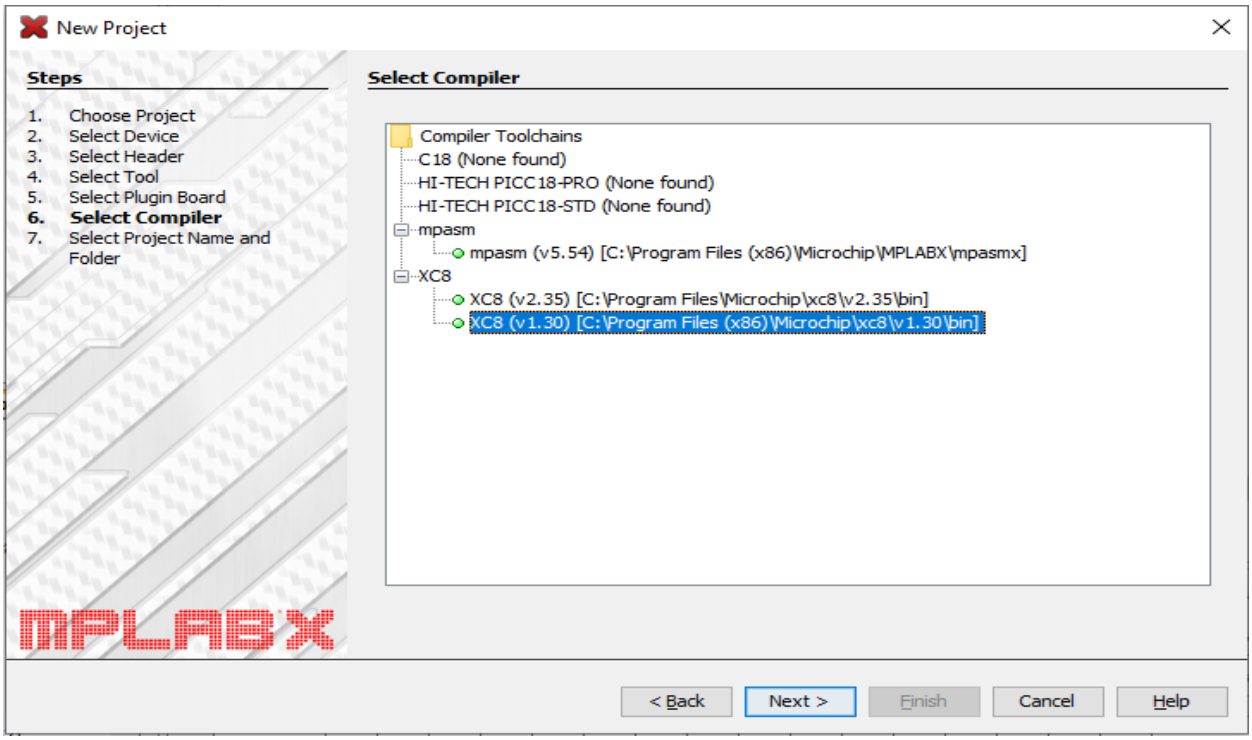
→ Next → Select Device → Family → Select Advanced 8-bit MCUs (PIC18) → Device → PIC18F4550



**Step 3:- Select Tool** → Here we have to select Simulator

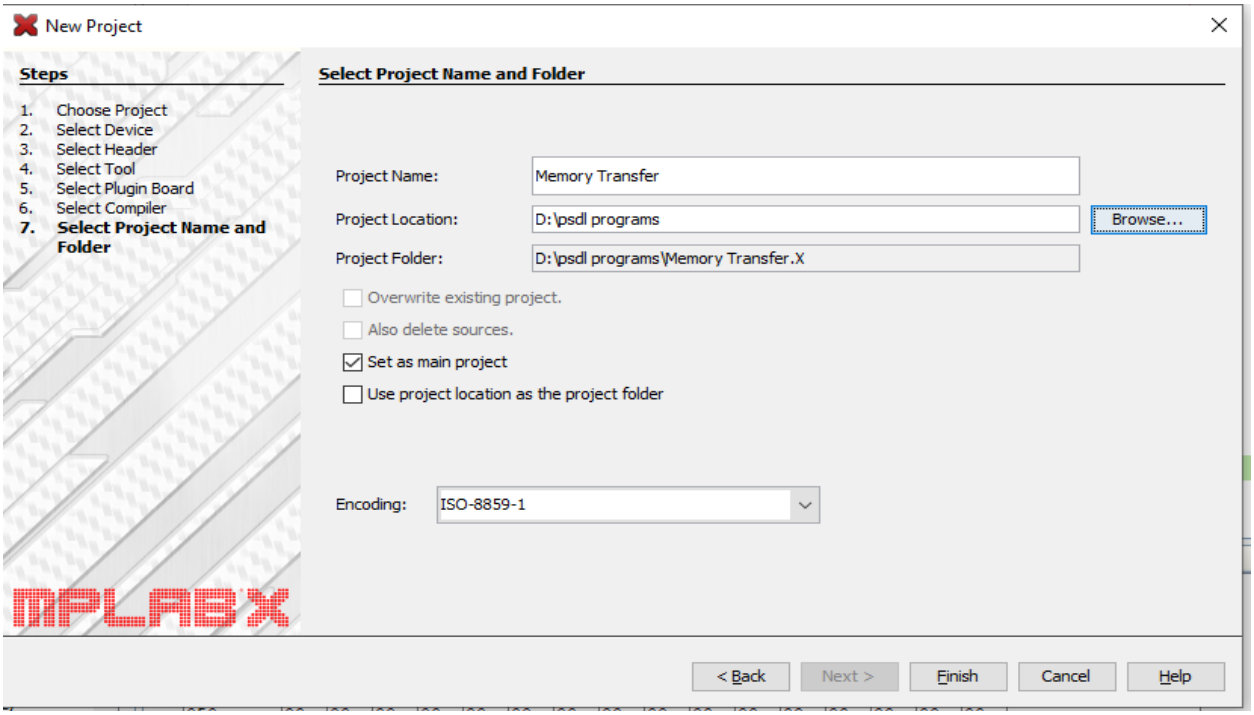


**Step 4 :- Select Compiler:-** Strictly we have to select the XC8 (v1.30) Compiler required for our project

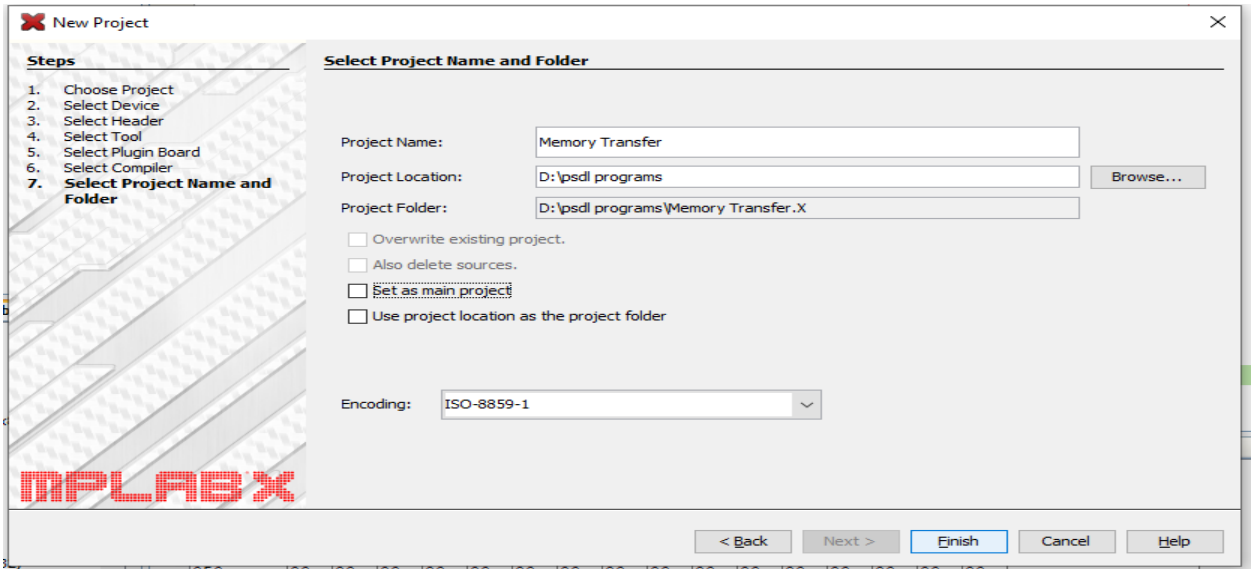


## Step 5 :- Select Project Name & Folder

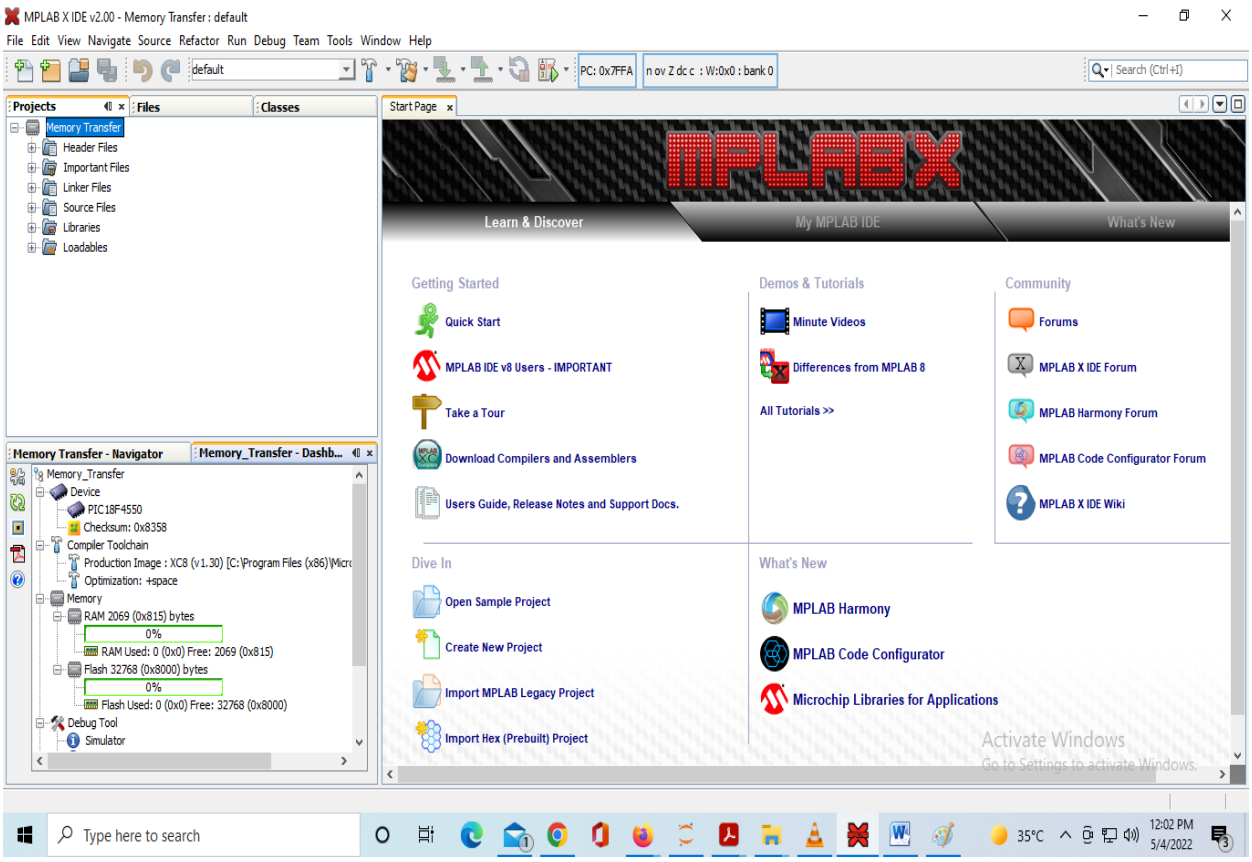
It is up to you to relevant name to the project and location to save project in your computer



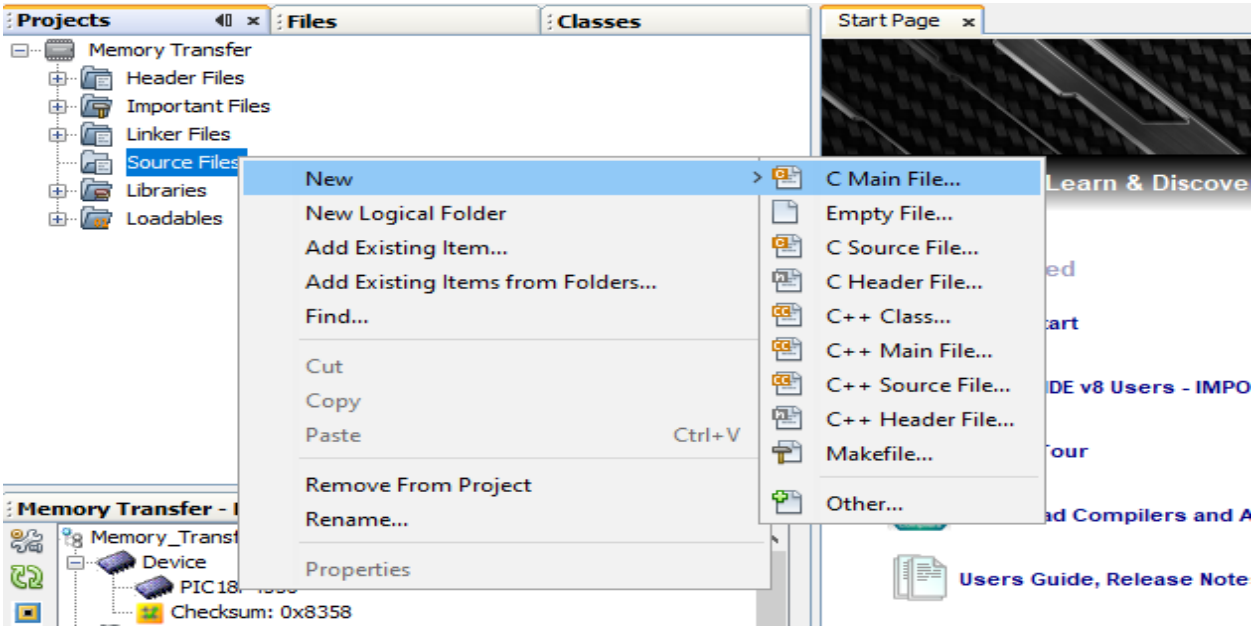

Deselect Set as main project

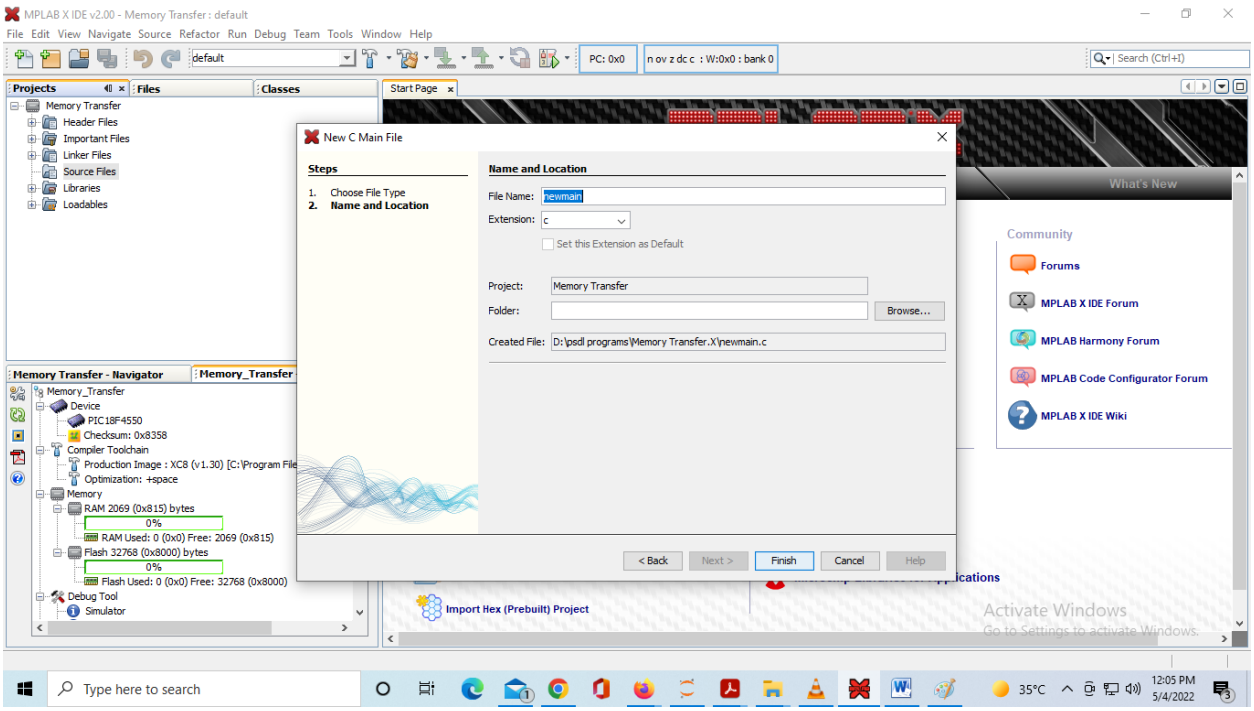


Now you can see new project is created




Now Right click on Source File → New → C Main File



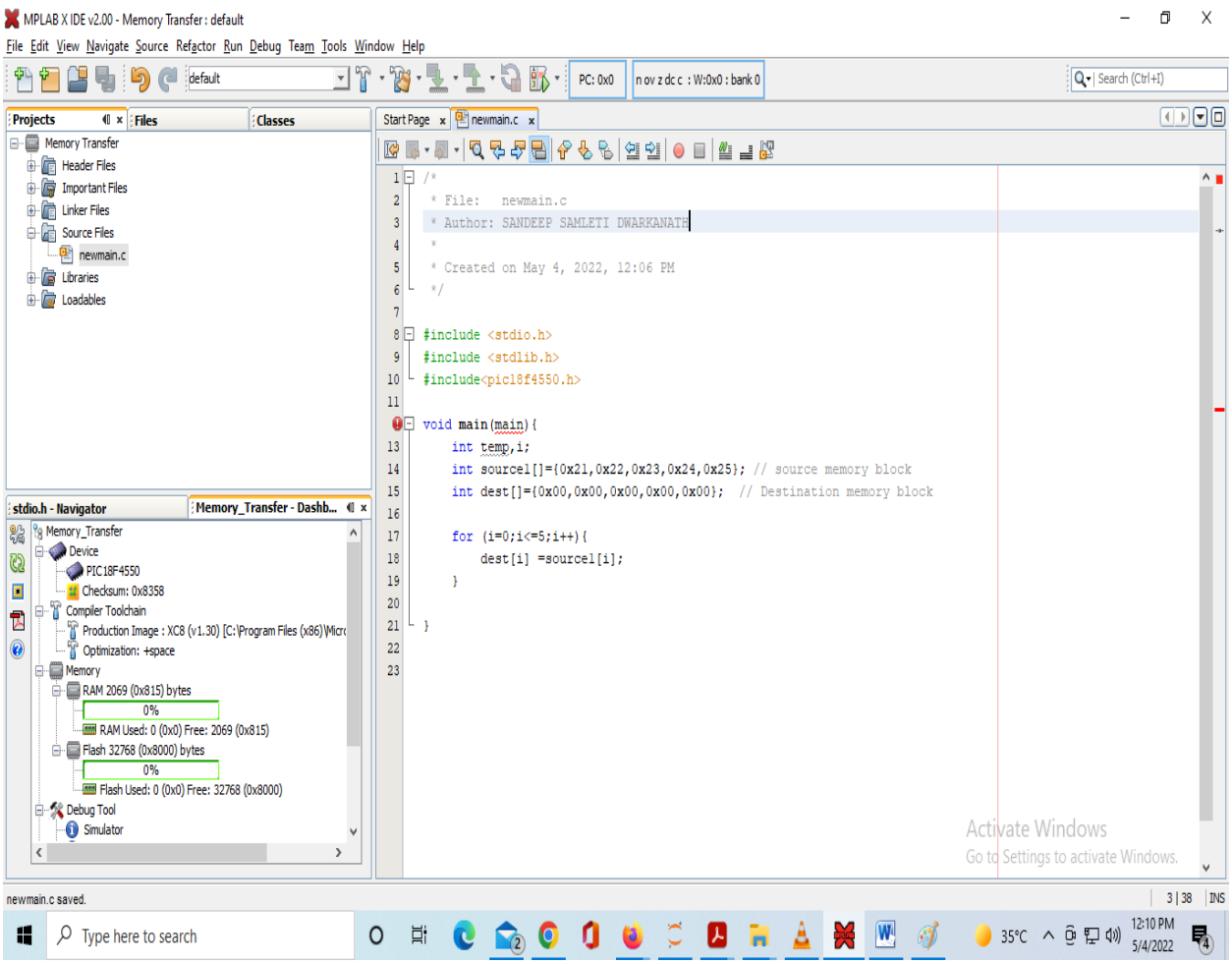



Now Click on Finish Button

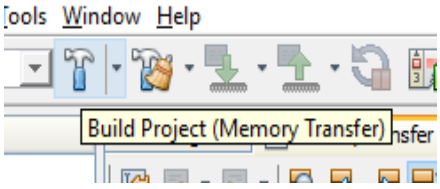
newmain.c file will be created in Source File

Now write the code in the code window

```
#include <stdio.h>
#include <stdlib.h>
#include<pic18f4550.h>
void main(main){
    int temp,i;
    int source1[]={0x21,0x22,0x23,0x24,0x25}; // source memory block
    int dest[]={0x00,0x00,0x00,0x00,0x00};  // Destination memory block
    for (i=0;i<=5;i++){
        dest[i] =source1[i];
    }  }
```

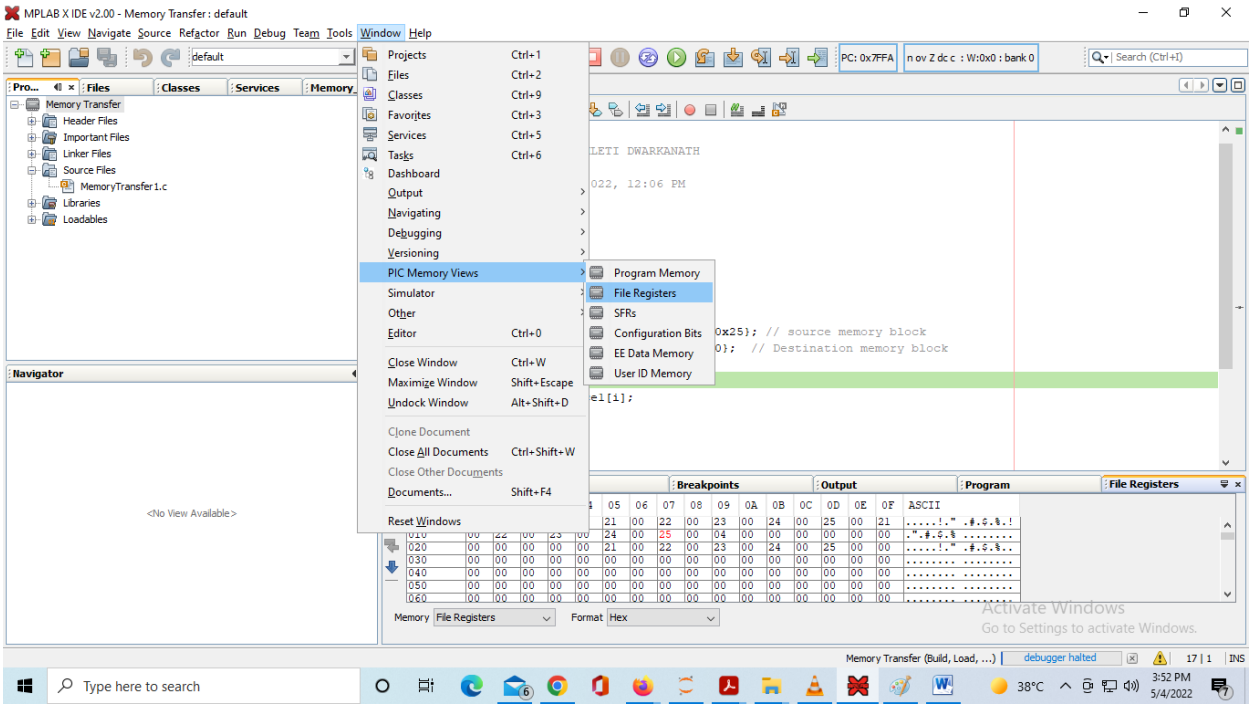



Now, Build the code

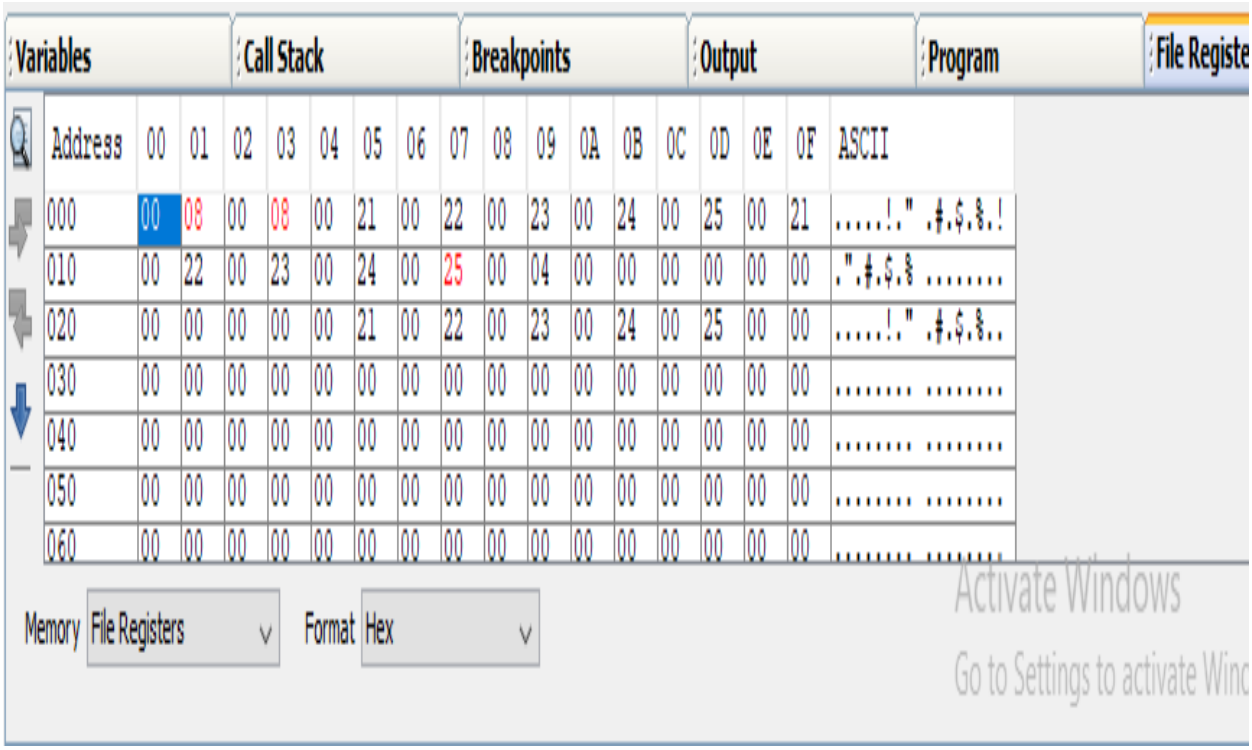
Click on Build Button 

This is the output we can see in the File Register

**To View Output Go to Window → PIC Memory Views → File Registers**

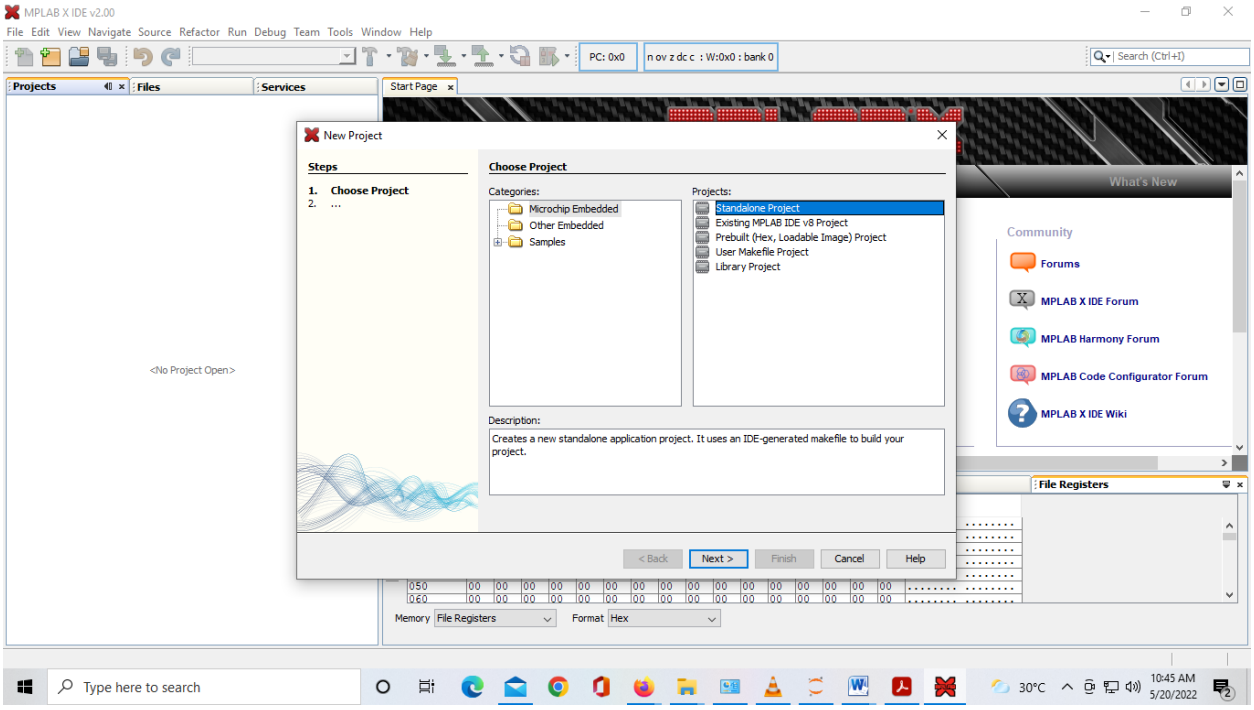



**Output:-**

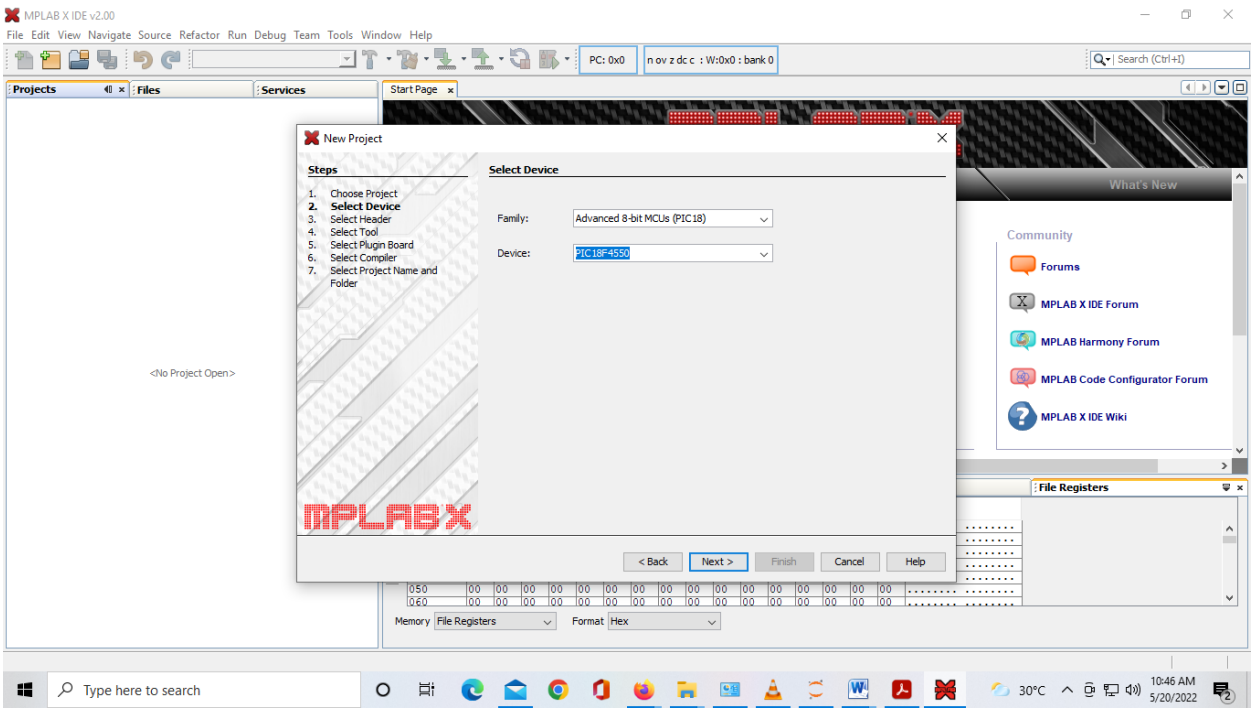


## Assignment No. 3 :- Multiplication of two numbers in PIC Microcontroller

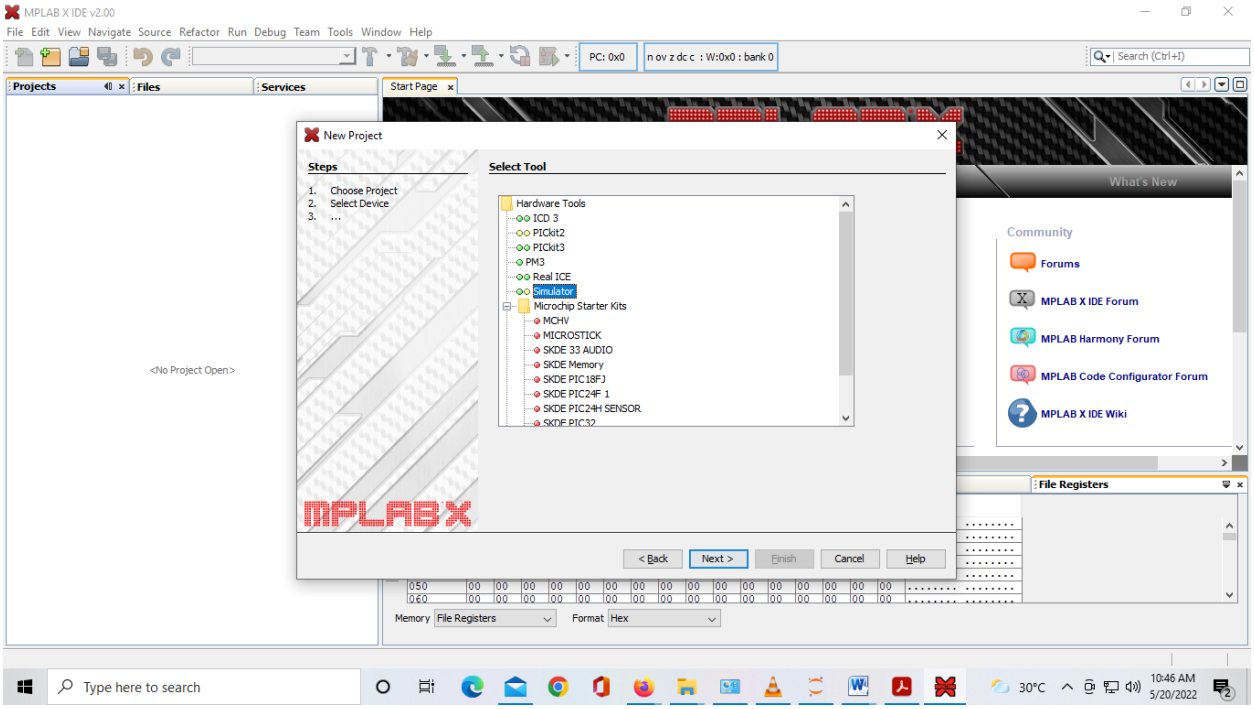
Step 1:- Select Project



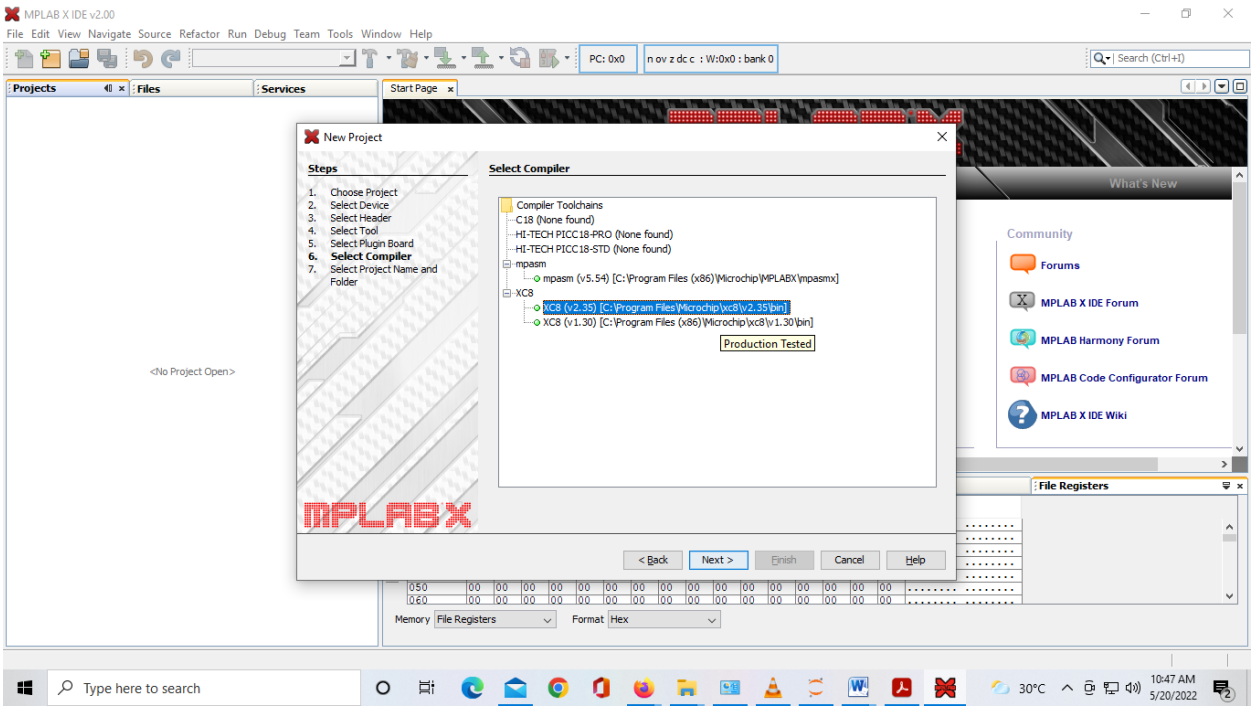

Select Device



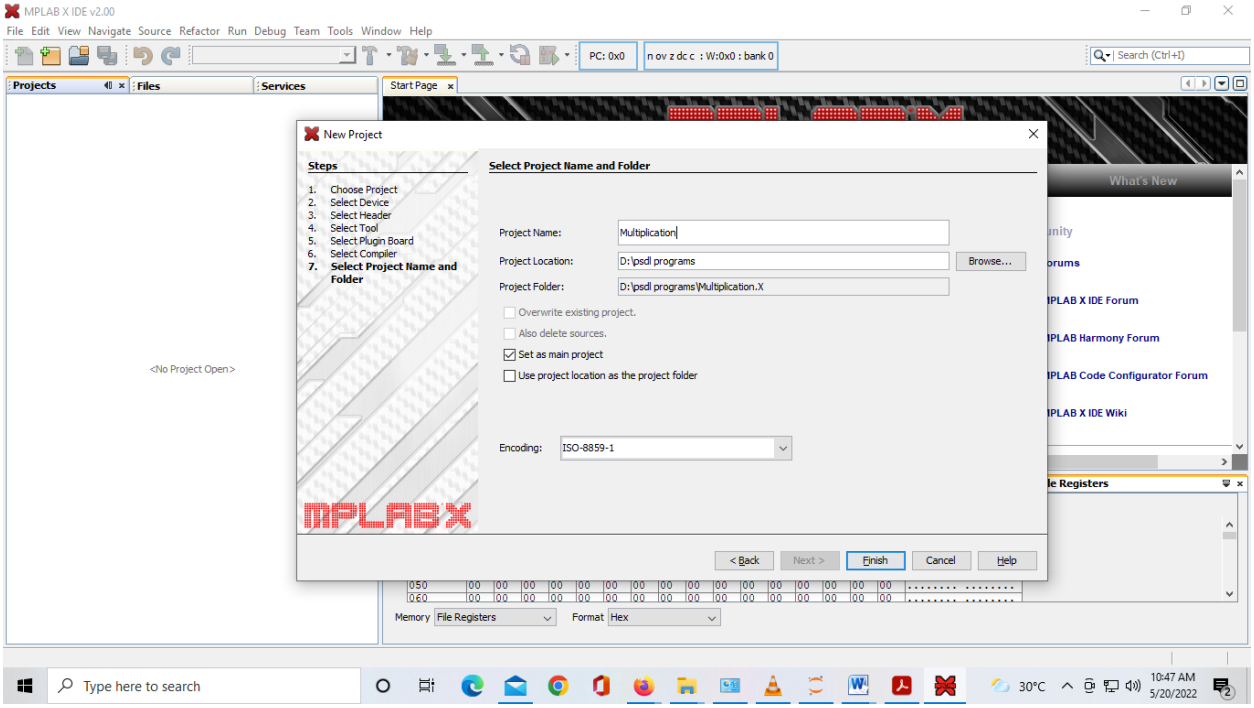
## Select Simulator




## Select Compiler



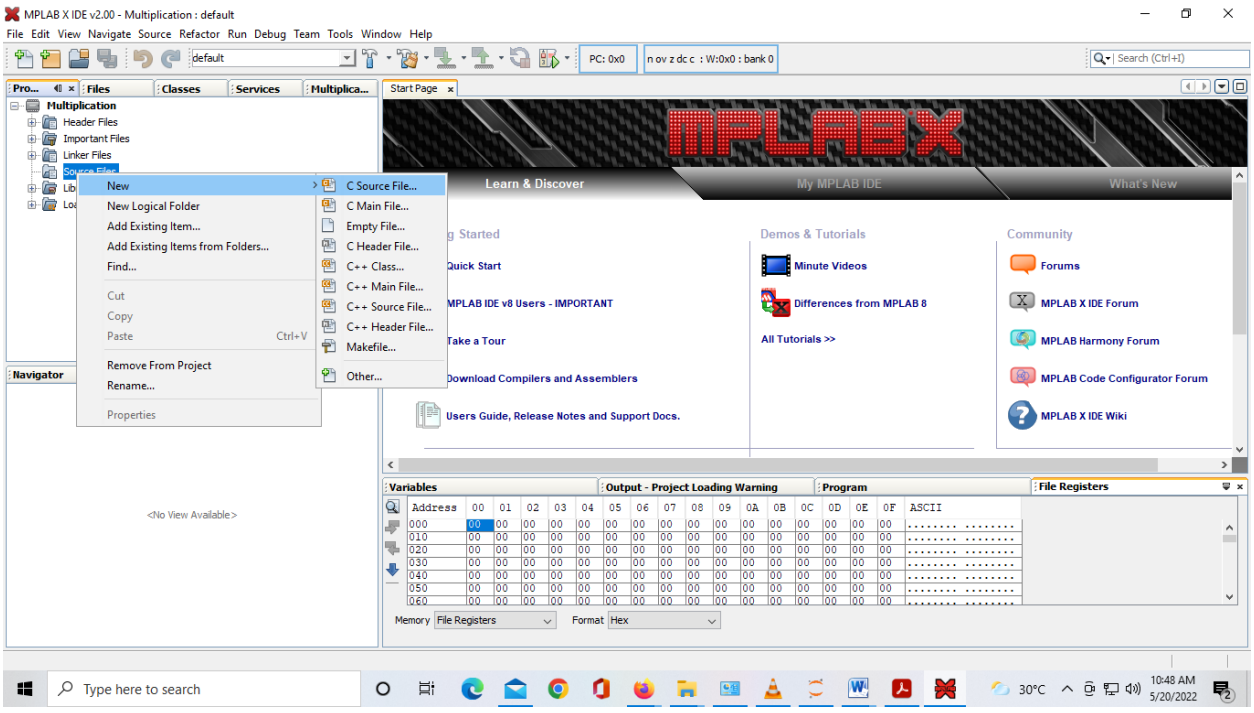
Project Name & Folder

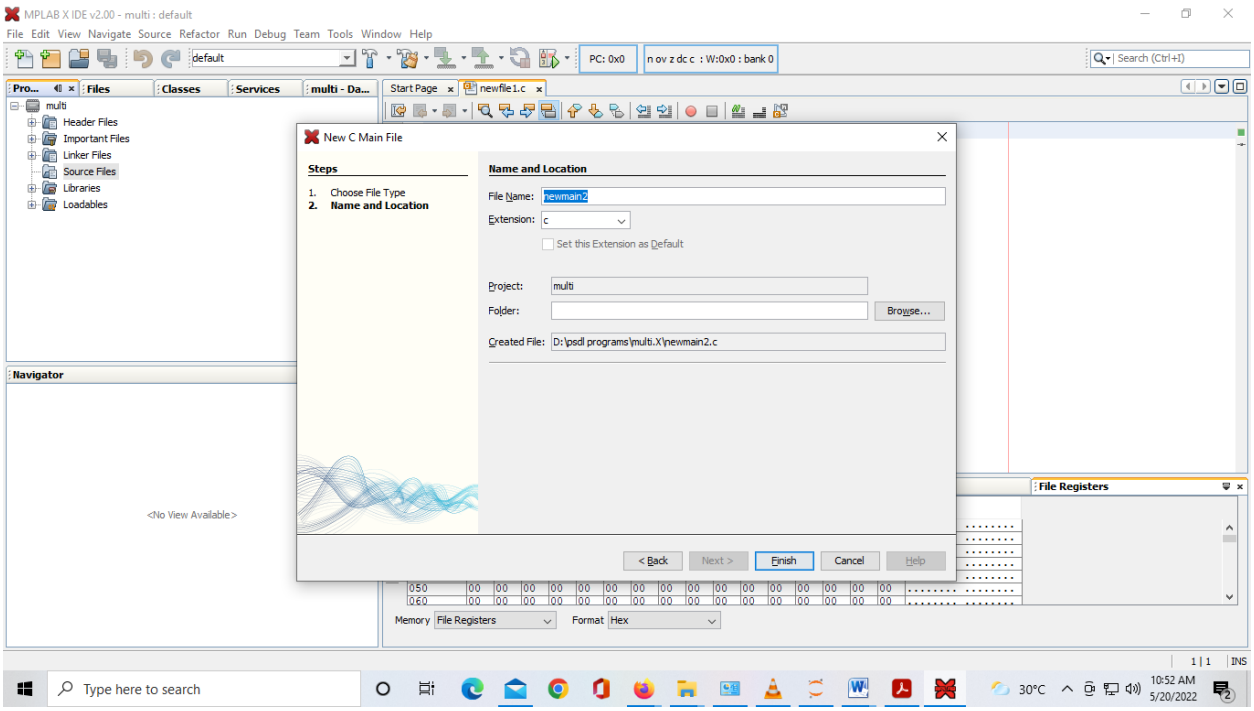


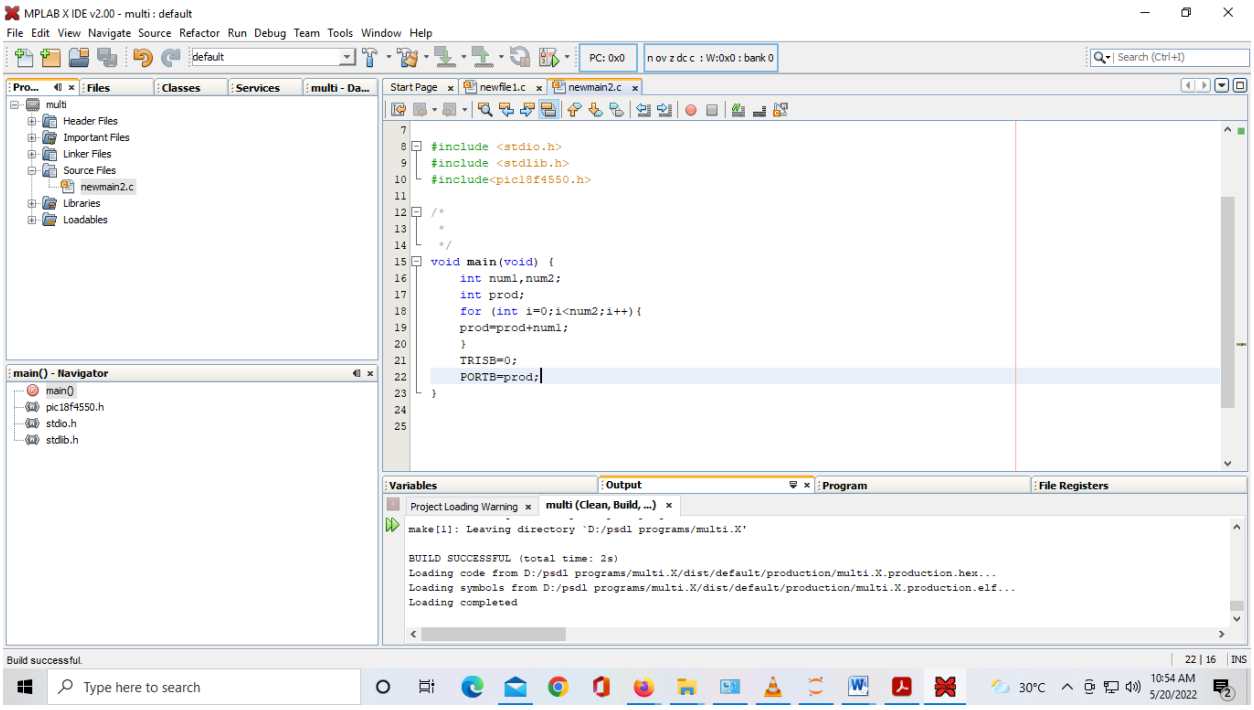

Add Source File :-

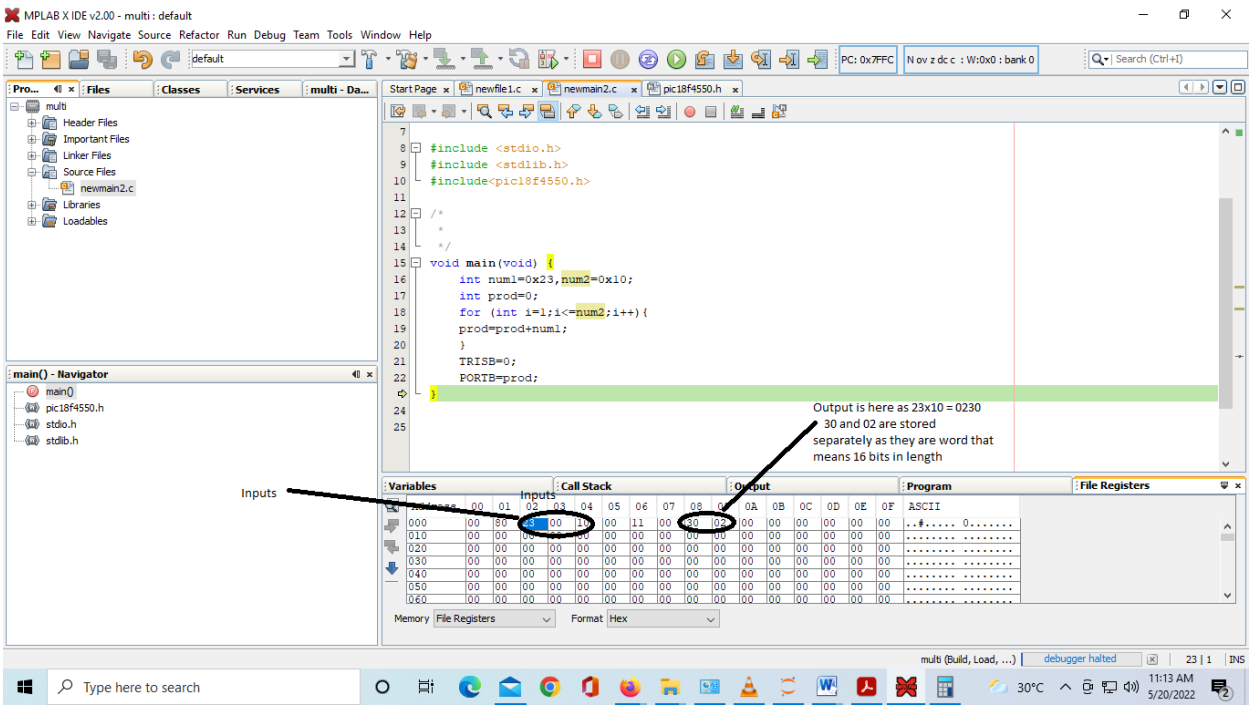
Right Click on Source Files

Select C Source File




MPLAB X IDE v2.00 - multi : default
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
New C Main File
Steps
1. Choose File Type
2. Name and Location
Name and Location
File Name: newmain2
Extension: c
Set this Extension as Default
Project: multi
Folder:
Browse...
Created File: D:\psdl programs\multi.X\newmain2.c
< Back
Next >
Finish
Cancel
Help
Navigator
<No View Available>
File Registers
Memory File Registers
Format Hex
Type here to search
30°C
10:52 AM
5/20/2022

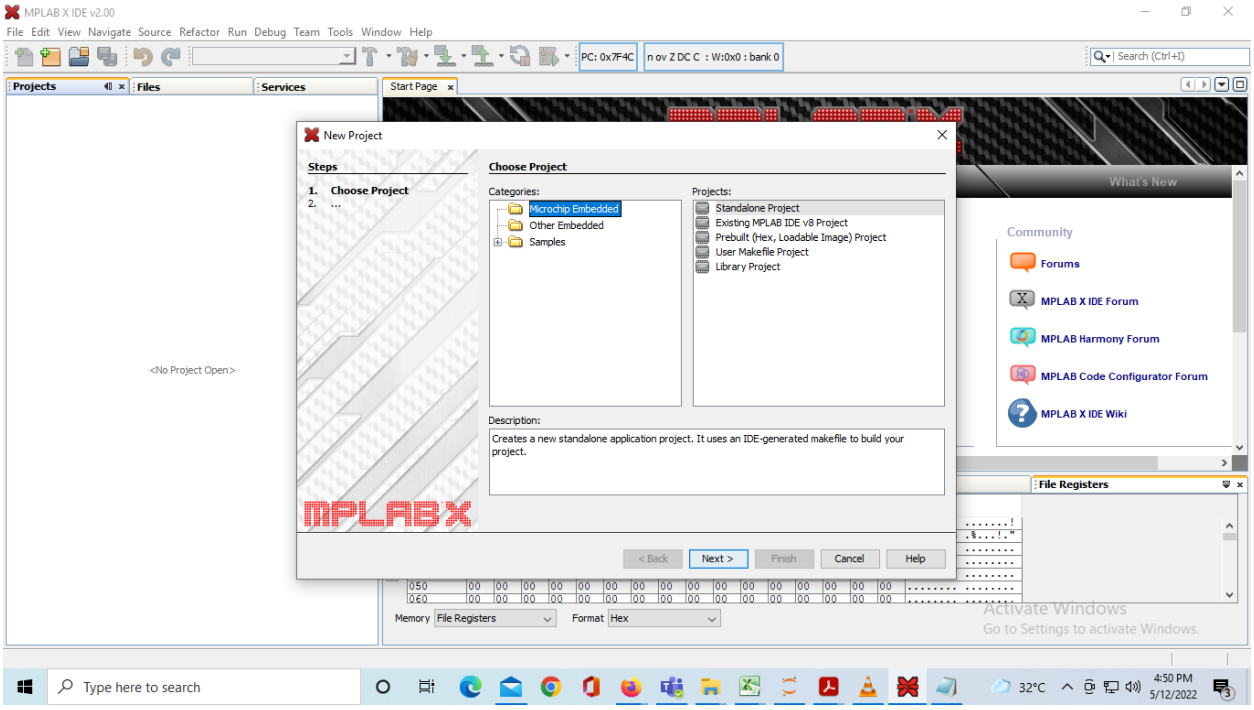

MPLAB X IDE v2.00 - multi : default
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
multi
Header Files
Important Files
Linker Files
Source Files
newmain2.c
Libraries
Loadables
main() - Navigator
main()
pic18f4550.h
stdio.h
stdlib.h
#include <stdio.h>
#include <stdlib.h>
#include<pic18f4550.h>
/*
*
*/
void main(void) {
int num1,num2;
int prod;
for (int i=0;i<num2;i++){
prod=prod+num1;
}
TRISB=0;
PORTB=prod;
}
Variables
Output
Program
File Registers
Project Loading Warning
multi (Clean, Build, ...)
make[1]: Leaving directory `D:/psdl programs/multi.X'
BUILD SUCCESSFUL (total time: 2s)
Loading code from D:/psdl programs/multi.X/dist/default/production/multi.X.production.hex...
Loading symbols from D:/psdl programs/multi.X/dist/default/production/multi.X.production.elf...
Loading completed
Build successful.
Type here to search
30°C
10:54 AM
5/20/2022

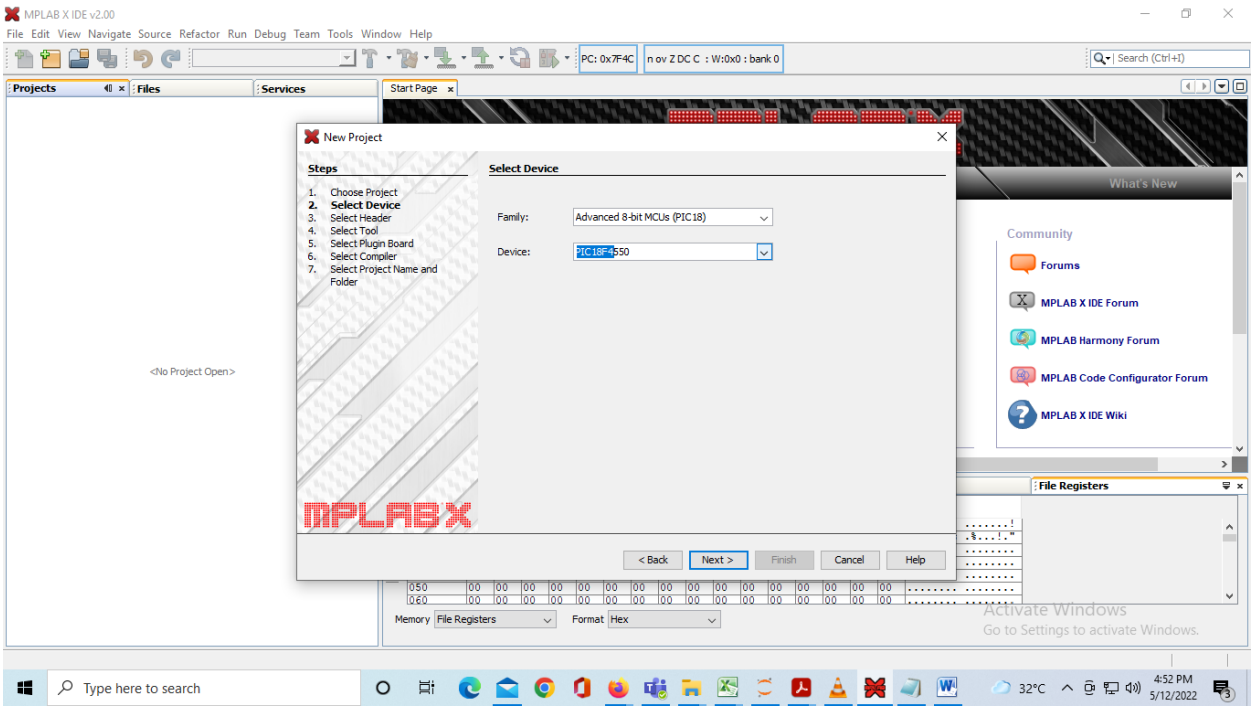


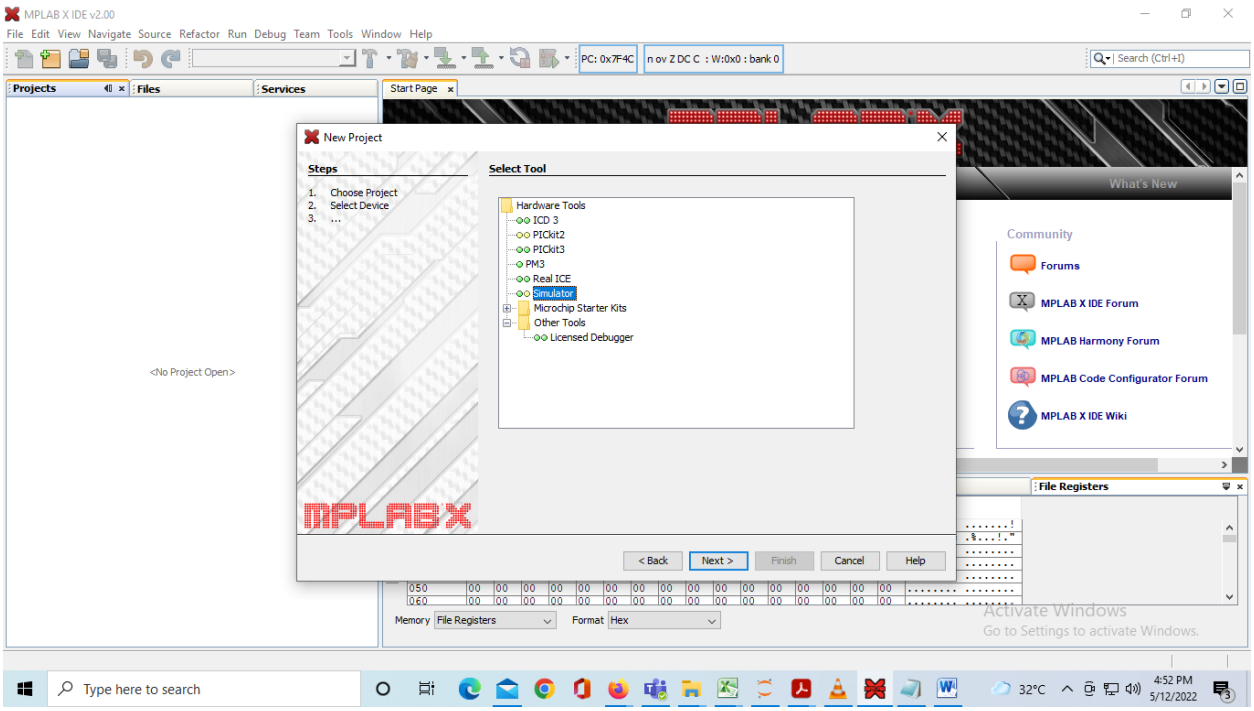

```
#include <stdio.h>
#include <stdlib.h>
#include<pic18f4550.h>


/*
 *
 */
void main(void) {
    int num1=0x23,num2=0x10;
    int prod=0;
    for (int i=1;i<=num2;i++){
    prod=prod+num1;
    }
    TRISB=0;
    PORTB=prod;
}
```

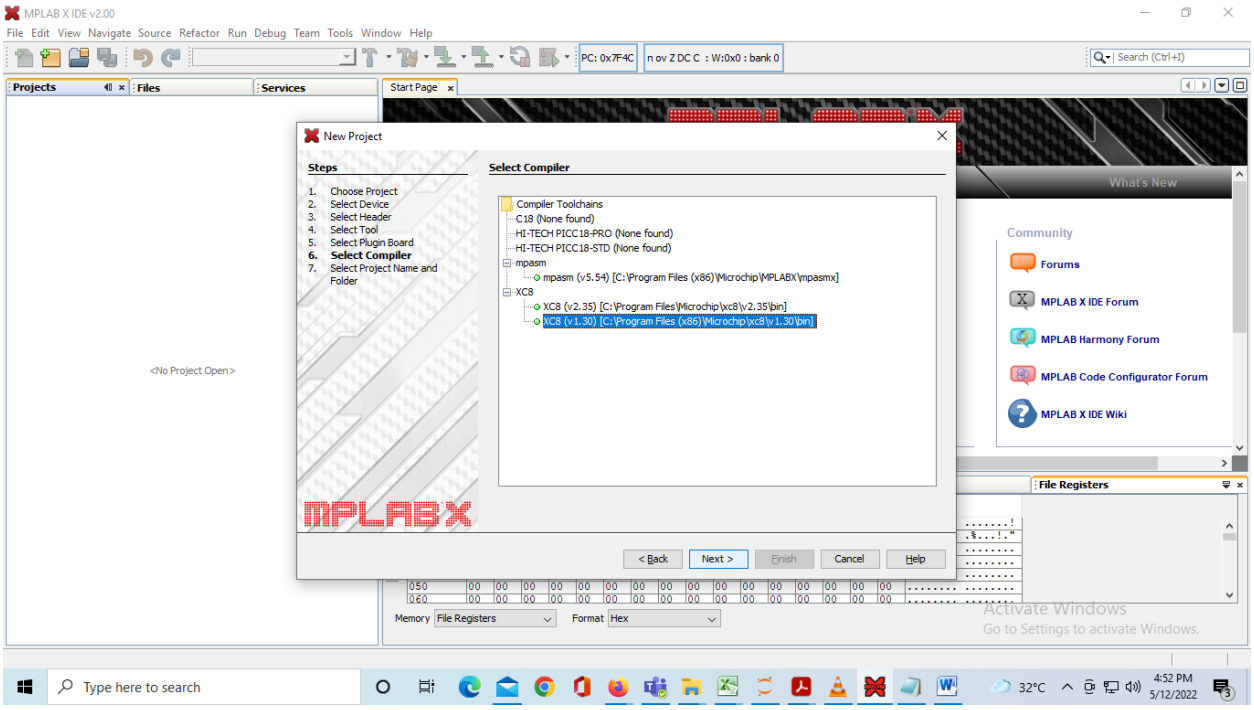
Assignment 4:- Memory Exchange

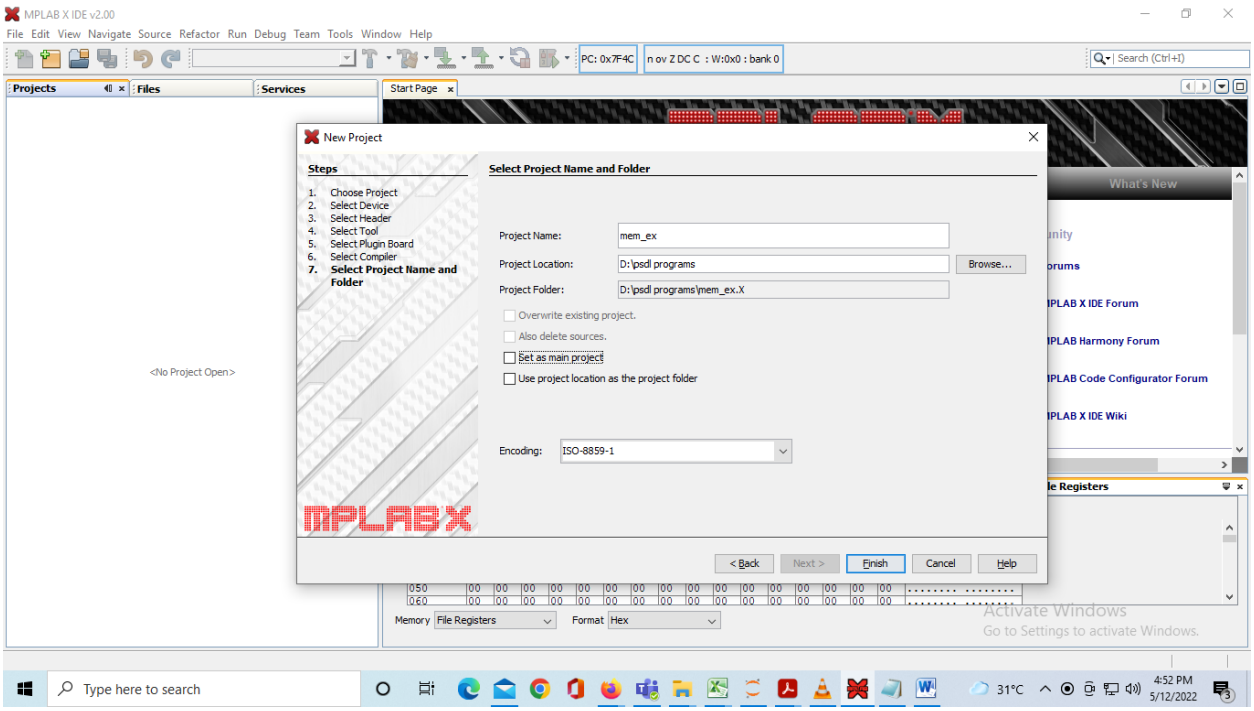
Step 1:- Device and Compiler Selection

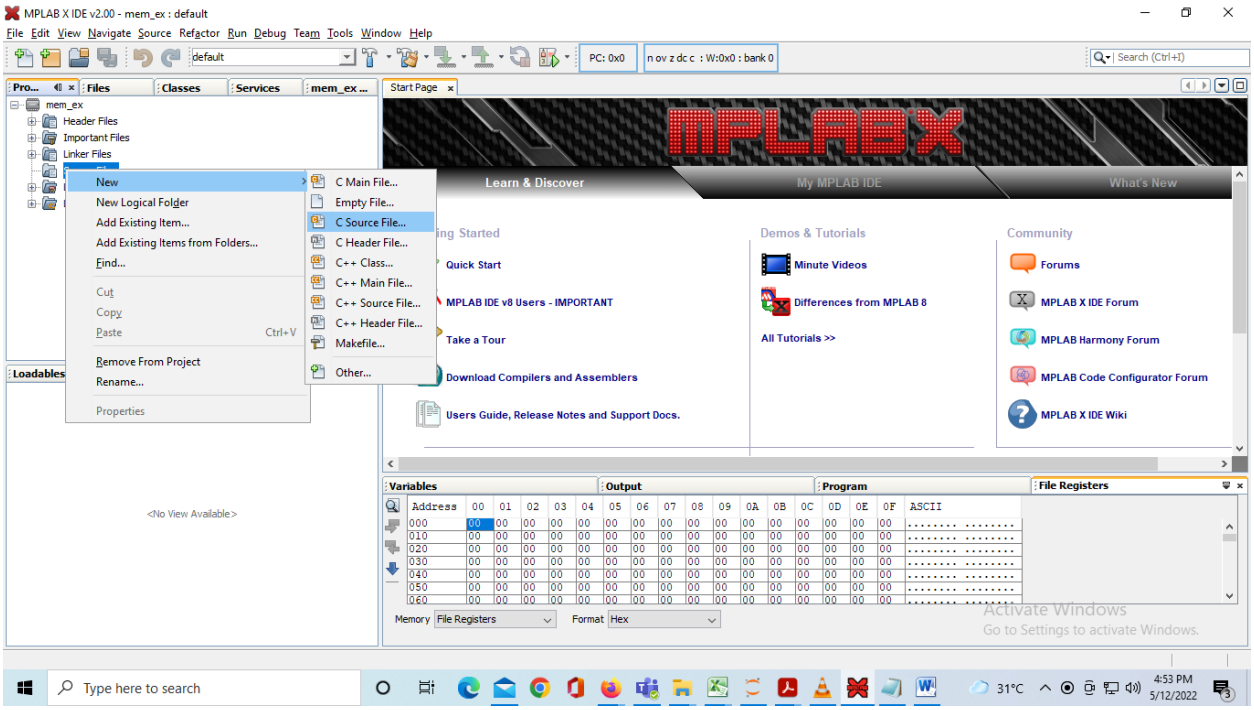


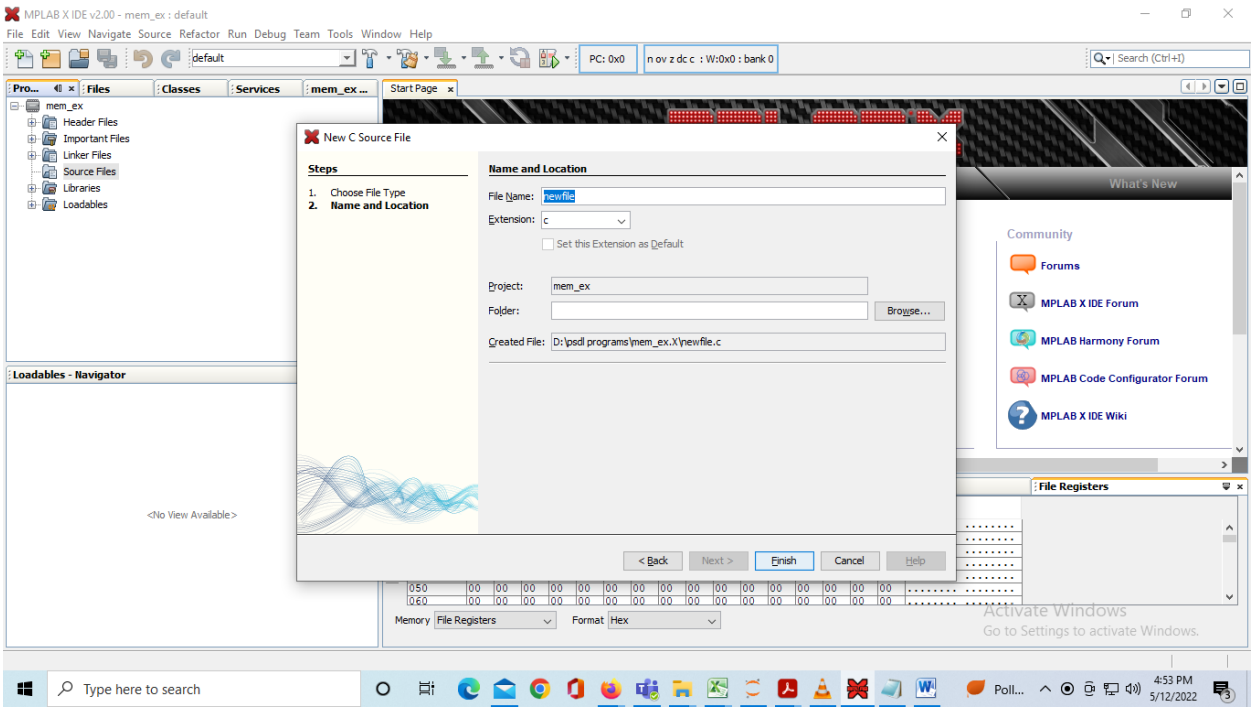




MPLAB X IDE v2.00
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Projects
Files
Services
Start Page
New Project
Steps
1. Choose Project
2. Select Device
3. ...
Select Tool
Hardware Tools
ICD 3
PICkit2
PICkit3
PM3
Real ICE
Simulator
Microchip Starter Kits
Other Tools
Licensed Debugger
< Back
Next >
Finish
Cancel
Help
<No Project Open>
What's New
Community
Forums
MPLAB X IDE Forum
MPLAB Harmony Forum
MPLAB Code Configurator Forum
MPLAB X IDE Wiki
File Registers
Memory
File Registers
Format
Hex
Activate Windows
Go to Settings to activate Windows.
Type here to search
32°C
4:52 PM
5/12/2022


MPLAB X IDE v2.00
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Projects
Files
Services
Start Page
New Project
Steps
1. Choose Project
2. Select Device
3. Select Header
4. Select Tool
5. Select Plugin Board
6. Select Compiler
7. Select Project Name and Folder
Select Compiler
Compiler Toolchains
C18 (None found)
HI-TECH PICC18-PRO (None found)
HI-TECH PICC18-STD (None found)
mpasm
mpasm (v5.54) [C:\Program Files (x86)\Microchip\MPLABX\mpasmx]
XC8
XC8 (v2.35) [C:\Program Files\Microchip\xc8\v2.35\bin]
XC8 (v1.30) [C:\Program Files (x86)\Microchip\xc8\v1.30\bin]
< Back
Next >
Finish
Cancel
Help
<No Project Open>
What's New
Community
Forums
MPLAB X IDE Forum
MPLAB Harmony Forum
MPLAB Code Configurator Forum
MPLAB X IDE Wiki
File Registers
Memory
File Registers
Format
Hex
Activate Windows
Go to Settings to activate Windows.
Type here to search
32°C
4:52 PM
5/12/2022


MPLAB X IDE v2.00
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Projects
Files
Services
Start Page
New Project
Steps
1. Choose Project
2. Select Device
3. Select Header
4. Select Tool
5. Select Plugin Board
6. Select Compiler
7. Select Project Name and Folder
Select Project Name and Folder
Project Name: mem_ex
Project Location: D:\psdl programs
Browse...
Project Folder: D:\psdl programs\mem_ex.X
Overwrite existing project.
Also delete sources.
Set as main project
Use project location as the project folder
Encoding: ISO-8859-1
< Back
Next >
Finish
Cancel
Help
<No Project Open>
What's New
Memory File Registers
Format Hex
Activate Windows
Go to Settings to activate Windows.
Type here to search
31°C
4:52 PM
5/12/2022

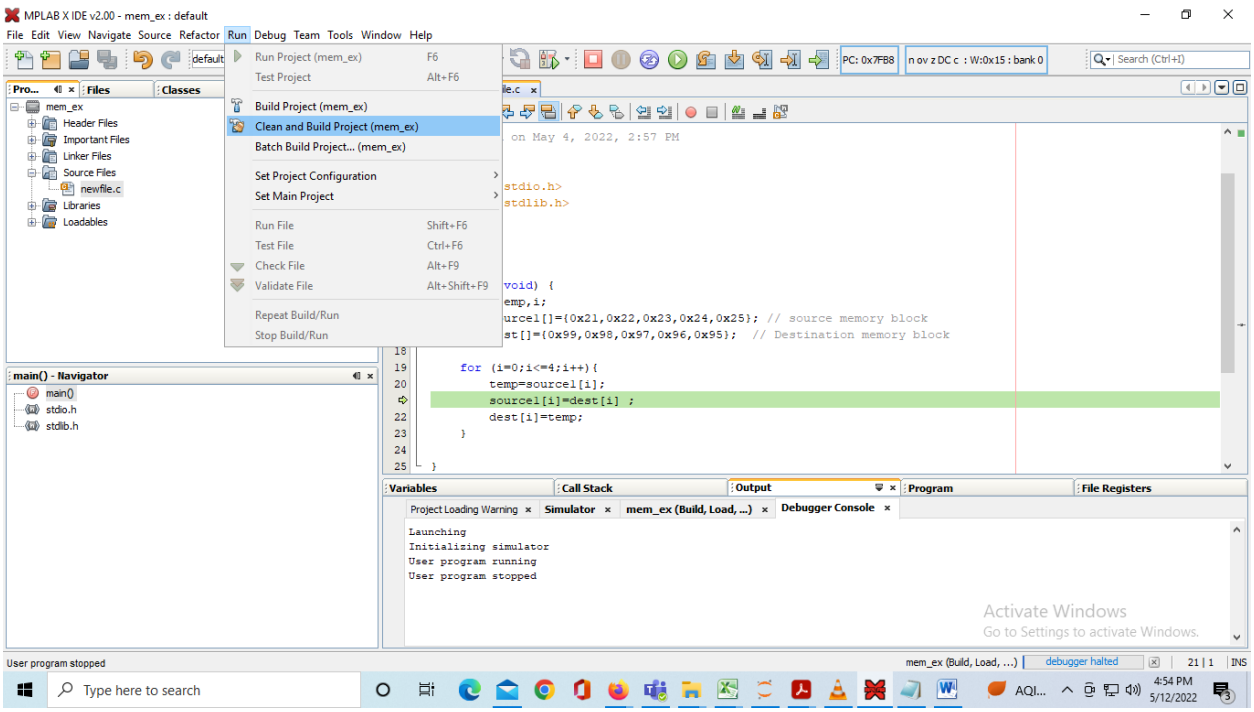

MPLAB X IDE v2.00 - mem_ex : default
File Edit View Navigate Source Refactor Run Debug Team Tools Window Help
Projects
Files
Classes
Services
mem_ex
Header Files
Important Files
Linker Files
New
New Logical Folder
Add Existing Item...
Add Existing Items from Folders...
Find...
Cut
Copy
Paste Ctrl+V
Remove From Project
Rename...
Properties
C Main File...
Empty File...
C Source File...
C Header File...
C++ Class...
C++ Main File...
C++ Source File...
C++ Header File...
Makefile...
Other...
Loadables
<No View Available>
Start Page
Learn & Discover
My MPLAB IDE
What's New
Quick Start
MPLAB IDE v8 Users - IMPORTANT
Take a Tour
Download Compilers and Assemblers
Users Guide, Release Notes and Support Docs.
Demos & Tutorials
Minute Videos
Differences from MPLAB 8
All Tutorials >>
Community
Forums
MPLAB X IDE Forum
MPLAB Harmony Forum
MPLAB Code Configurator Forum
MPLAB X IDE Wiki
Variables
Output
Program
File Registers
Memory File Registers
Format Hex
Activate Windows
Go to Settings to activate Windows.
Type here to search
31°C
4:53 PM
5/12/2022

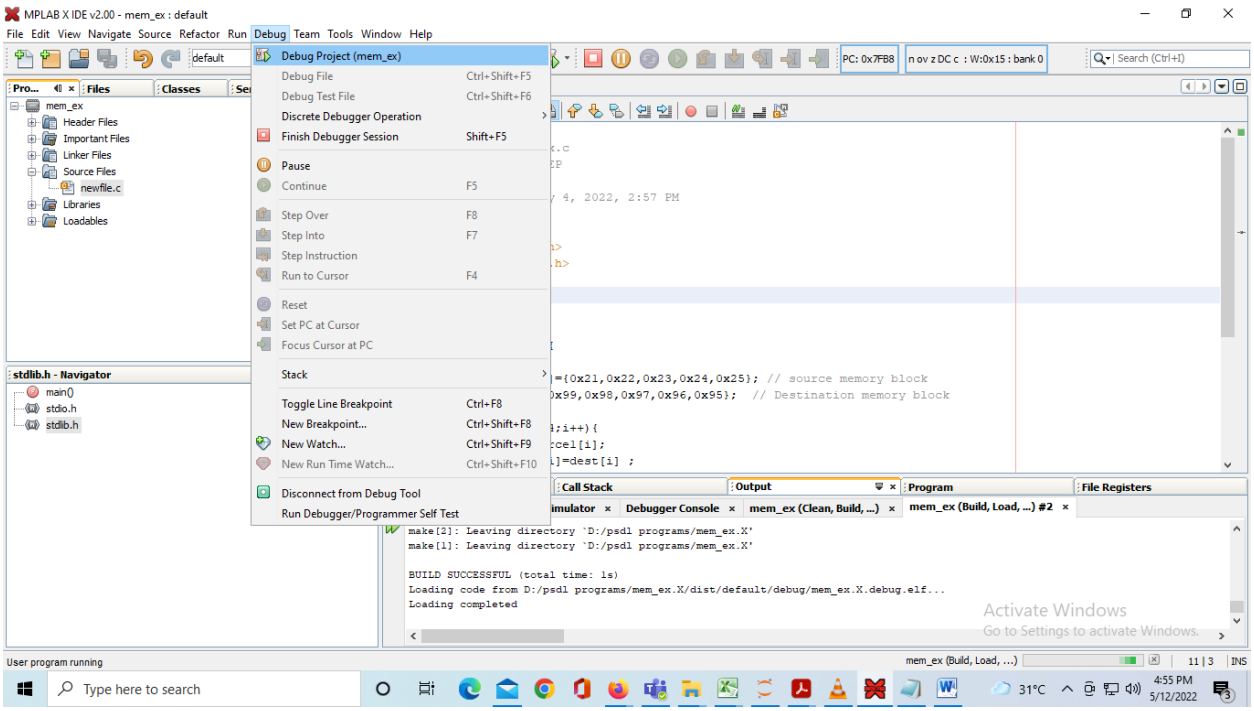



Copy the following code in the text window

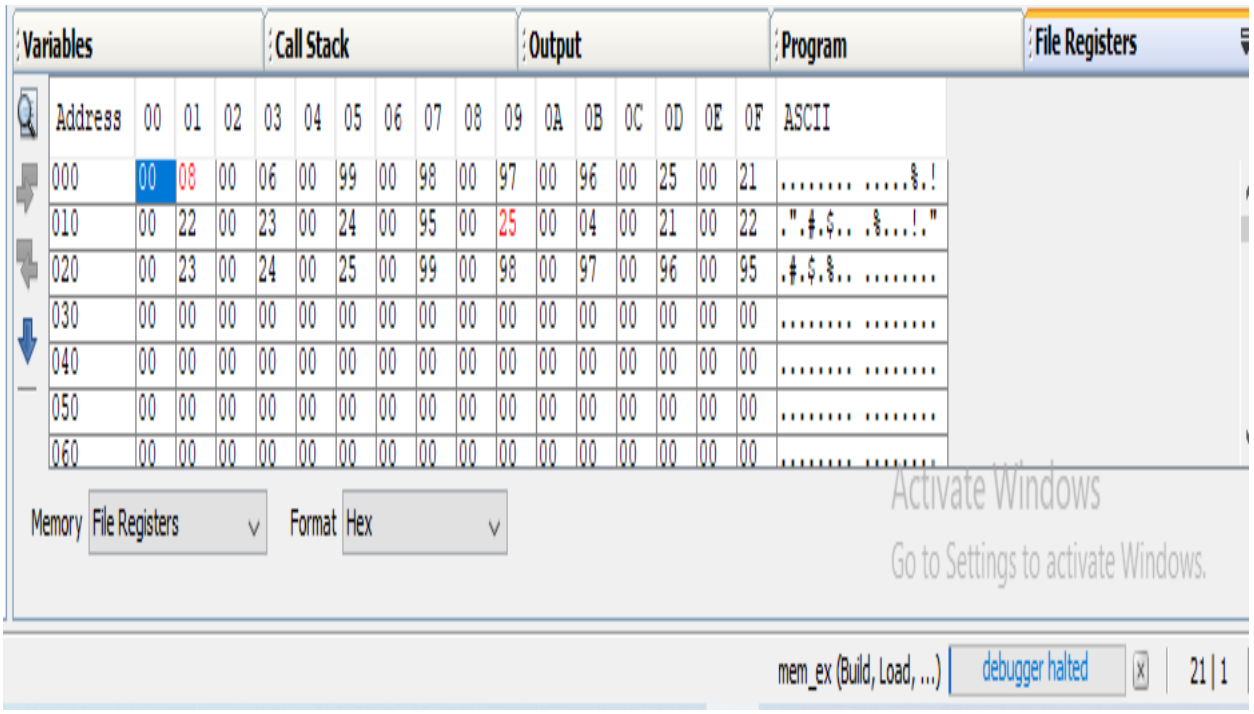
```
/*
 * File:   mem_ex.c
 * Author: SANDEEP
 *
 * Created on May 4, 2022, 2:57 PM
 */

#include <stdio.h>
#include <stdlib.h>

/*
 *
 */
void main(void) {
    int temp,i;
```

```
    int source1[]={0x21,0x22,0x23,0x24,0x25}; // source memory block
    int dest[]={0x99,0x98,0x97,0x96,0x95};  // Destination memory block

    for (i=0;i<=4;i++){
        temp=source1[i];
        source1[i]=dest[i] ;
        dest[i]=temp;
    }

}
```





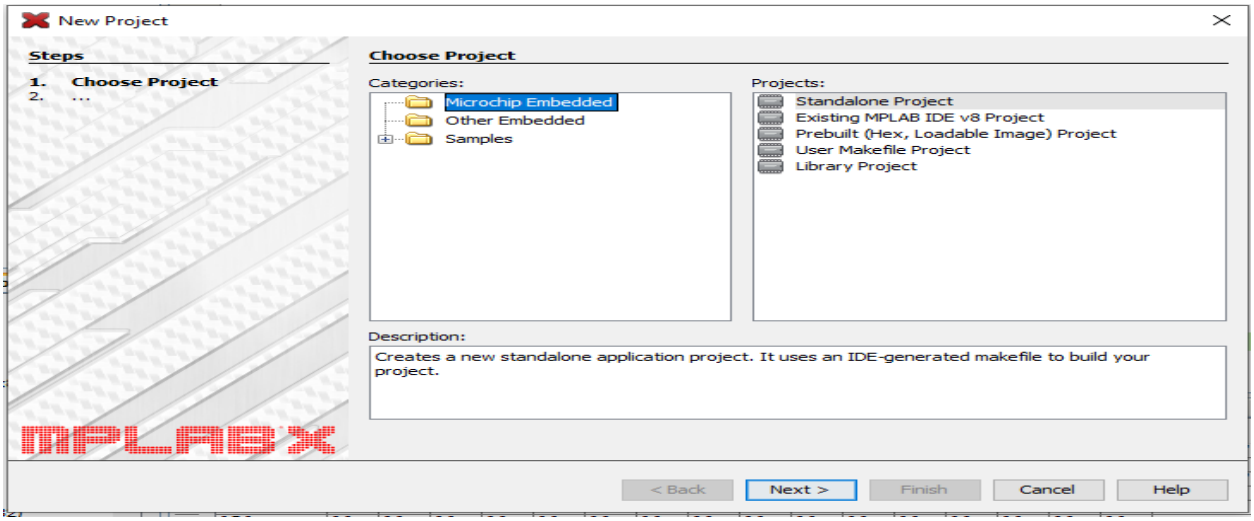

Output Can be seen here



**Assignment 5:- Sorting in  Ascending Order**

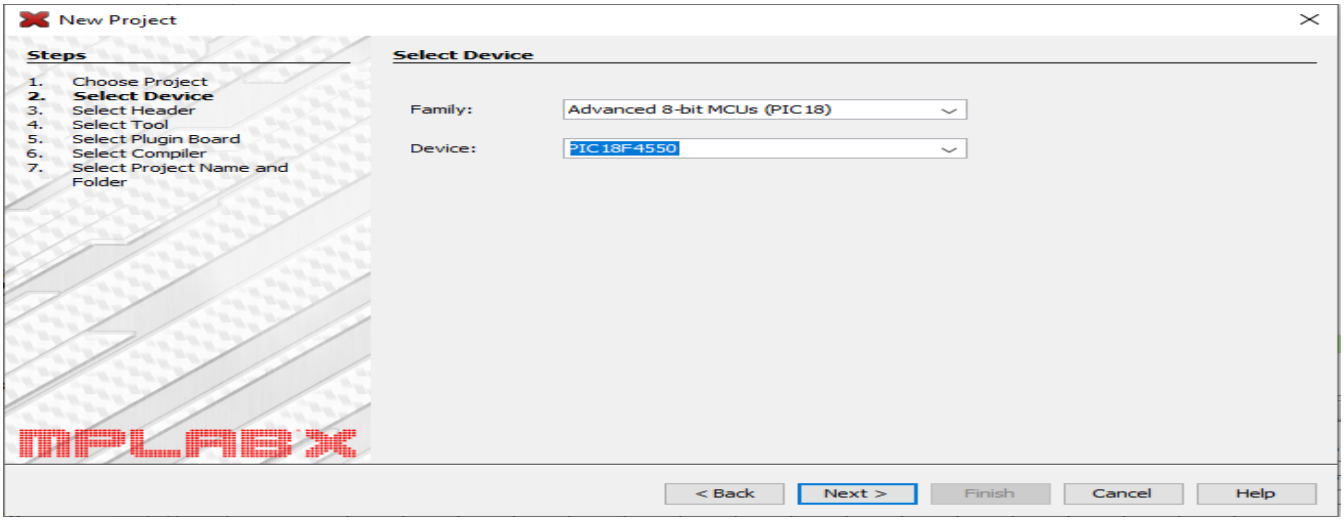
**Step 1 :-** Initial Configuration

File → New Project → Select Microchip Embedded → Standalone Project

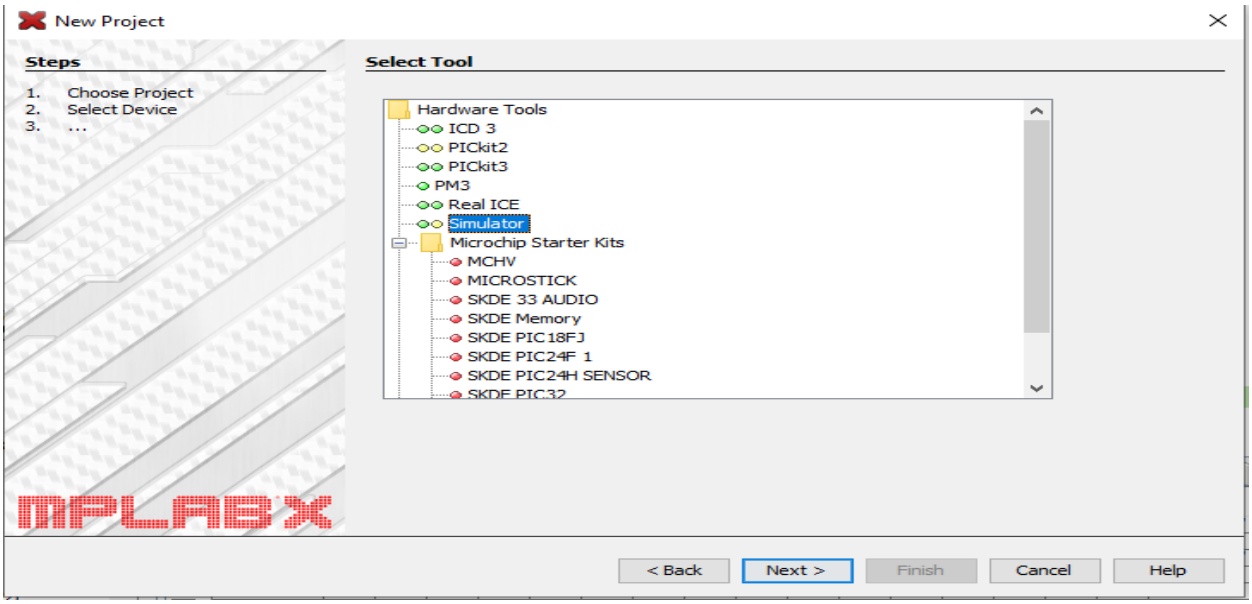


## Select Device

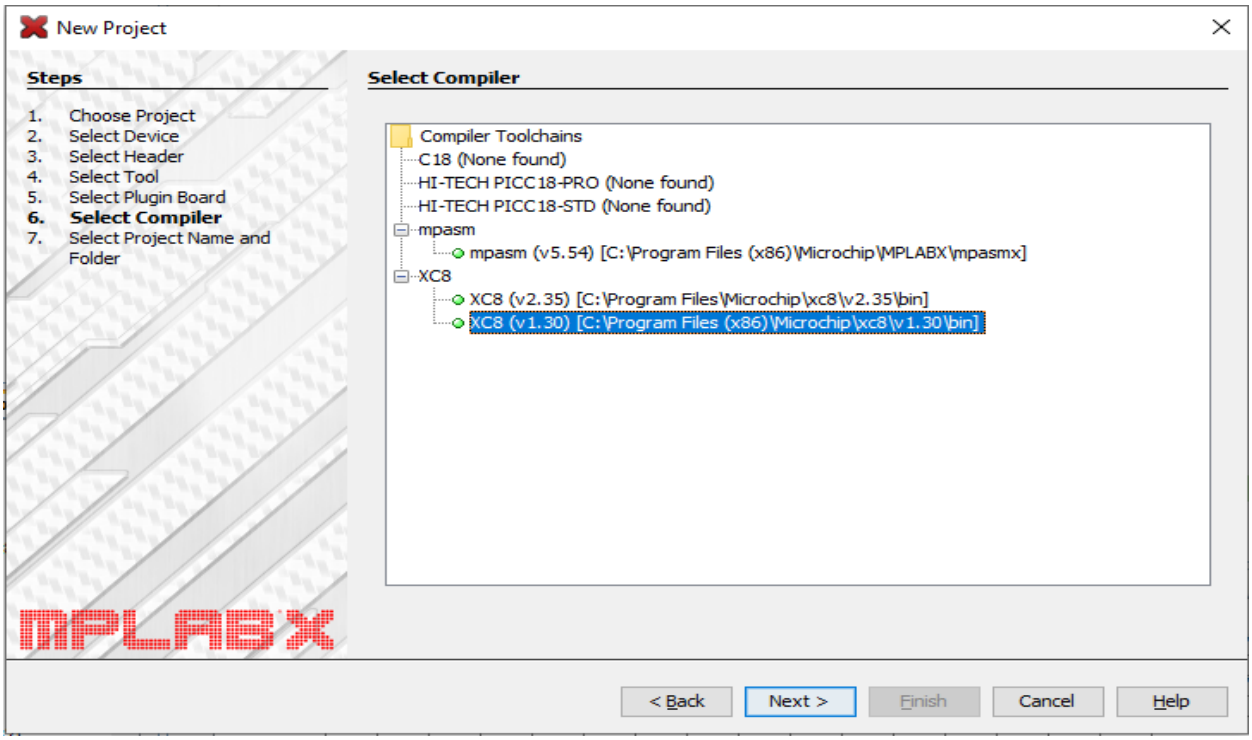
→ Next → Select Device → Family → Select Advanced 8-bit MCUs (PIC18) → Device → PIC18F4550



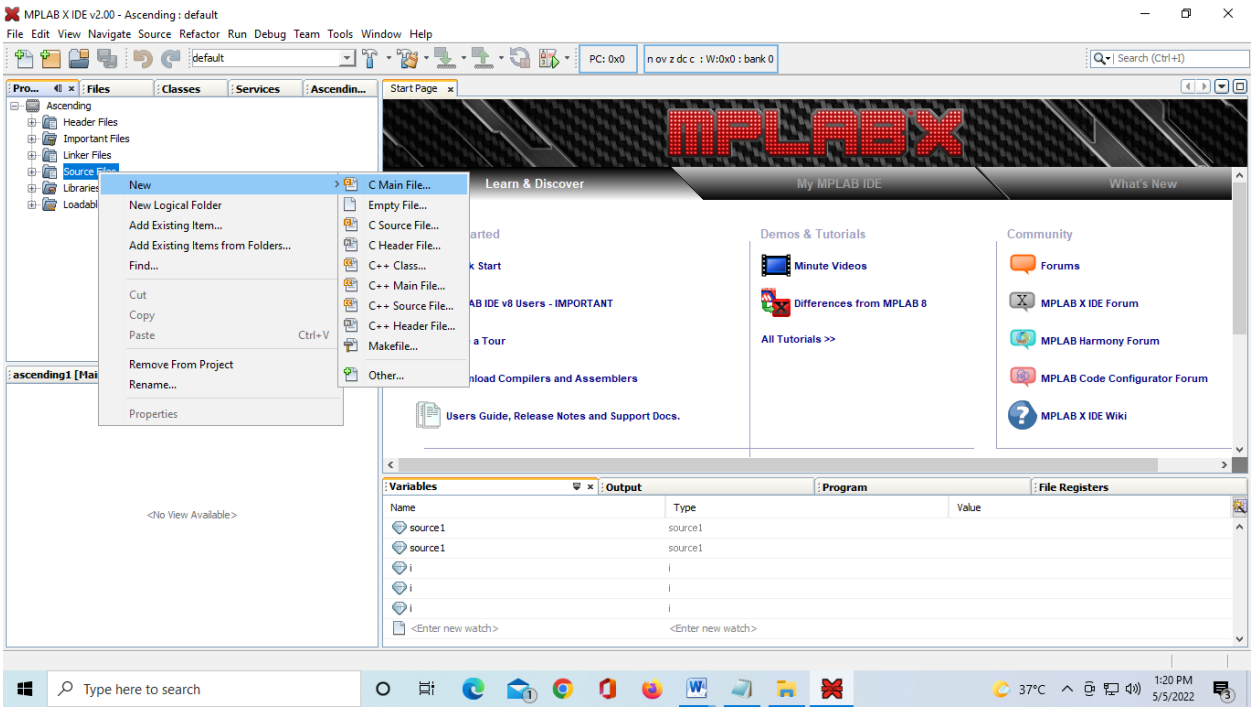

**Select Tool** → Here we have to select Simulator

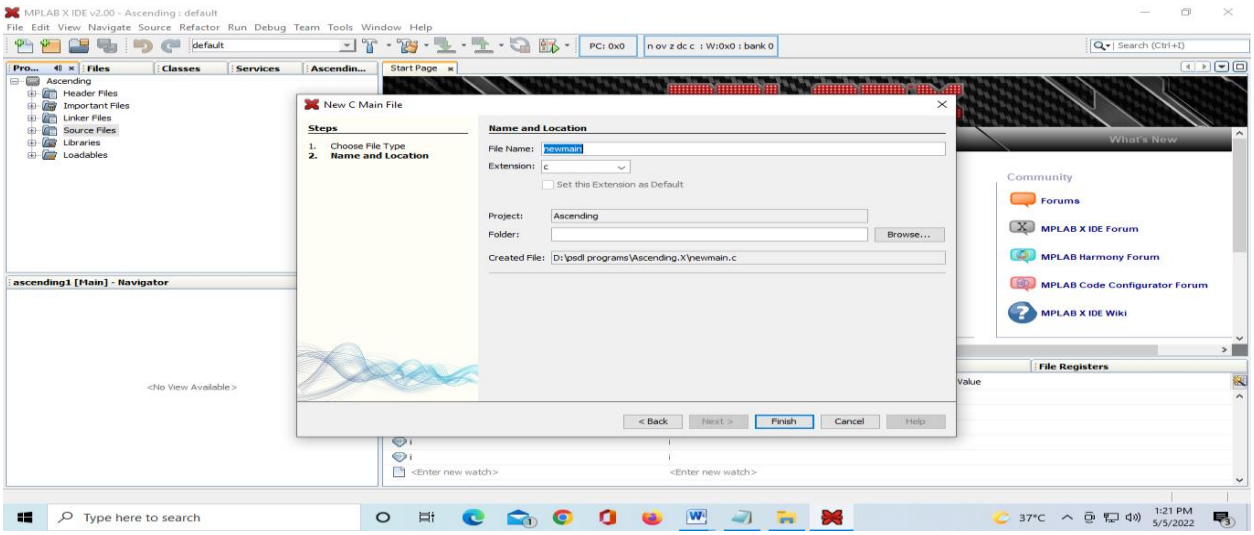


**Select Compiler:-** Strictly we have to select the XC8 (v1.30) Compiler required for our project

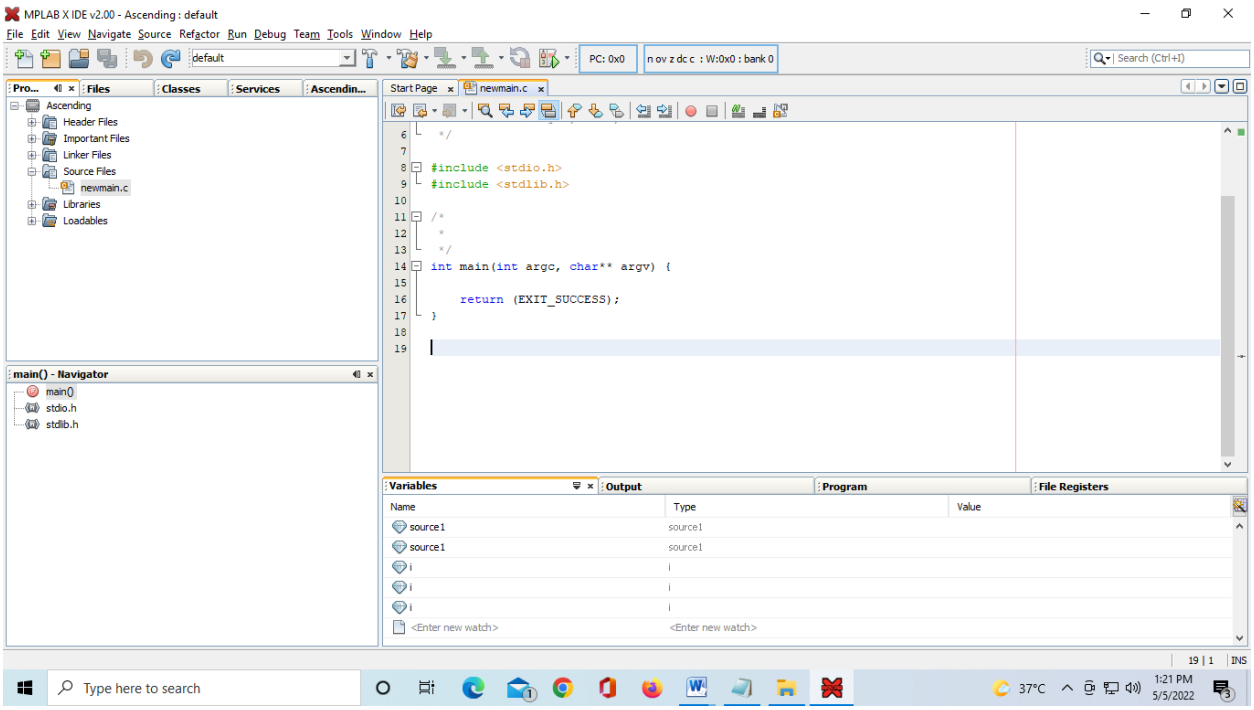



Step 2:- Right Click on Source File then select the New → C Main File





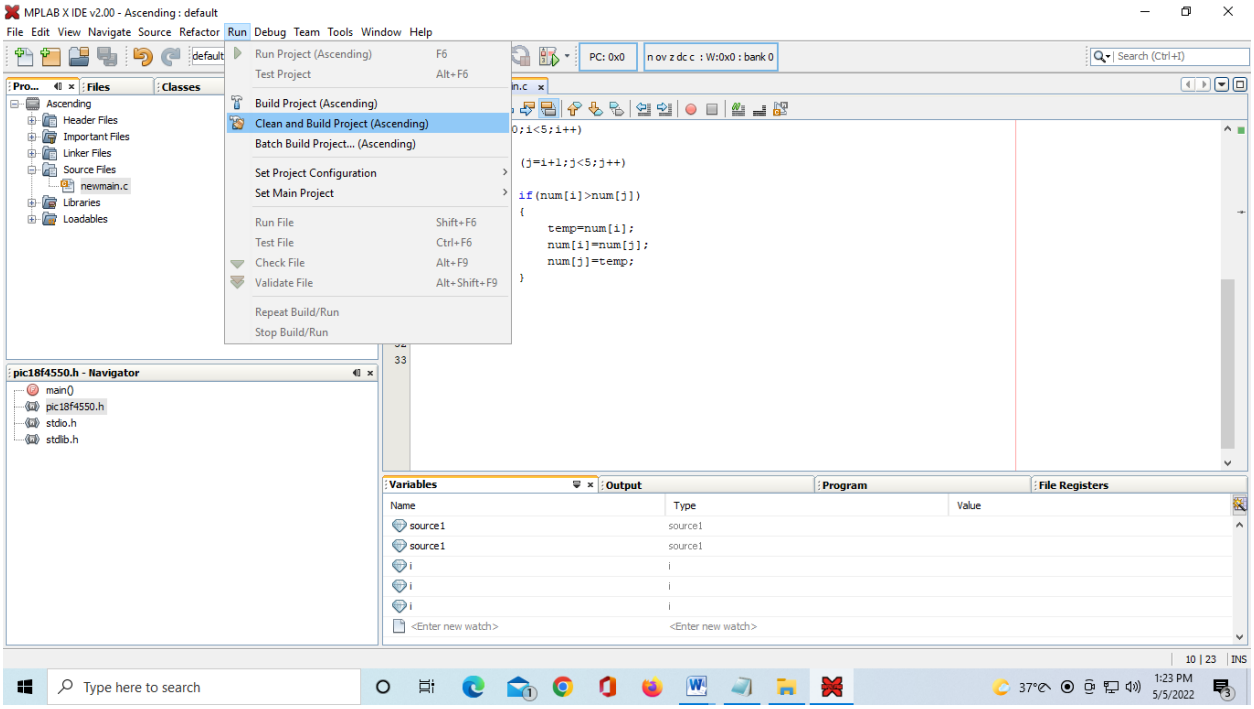

Type the following code in following code window



```
/*
 * File:   newmain1.c
 * Author: SANDEEP
 *
 * Created on May 5, 2022, 12:51 PM
 */
#include<pic18f4550.h>
#include <stdio.h>
#include <stdlib.h>
/*
 *
 */
void main(void)
{
    int temp,i,j;
    int num[]={10,2,5,1,6};
    for (i=0;i<5;i++)
    {
        for (j=i+1;j<5;j++)
        {
            if(num[i]>num[j])
            {
                temp=num[i];
                num[i]=num[j];
                num[j]=temp;
            }
        }
    }
}
```

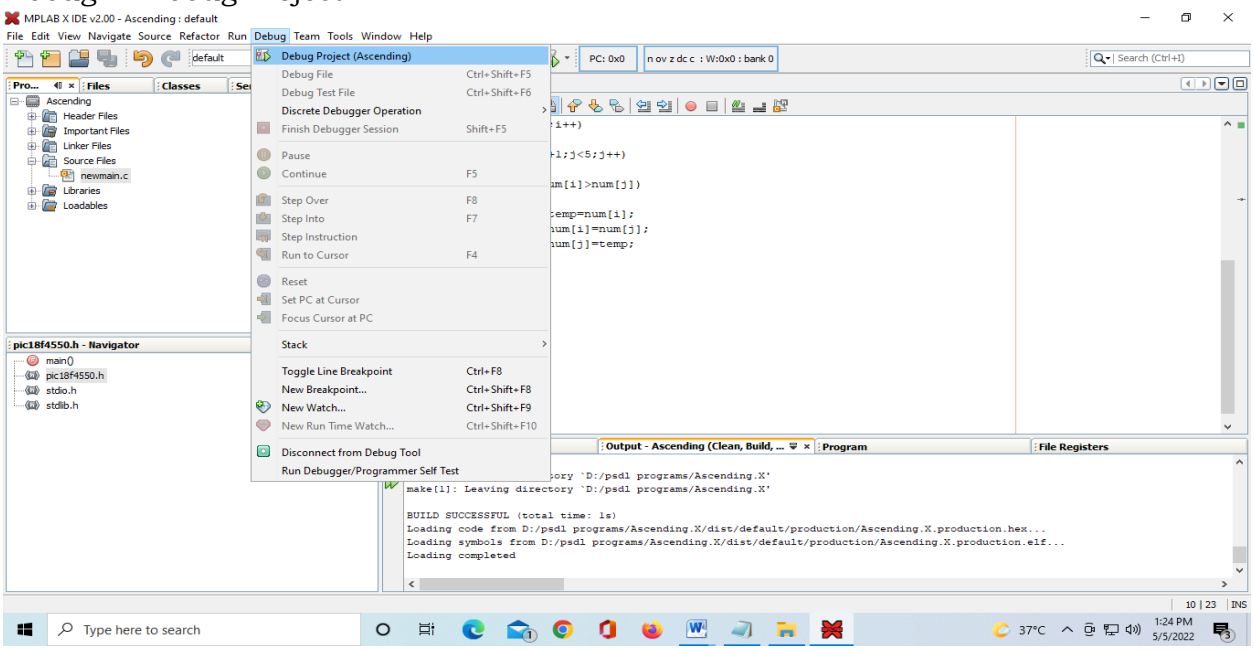
## Step 3:- Now Build the code

Under Run Menu Select → Clean and Build

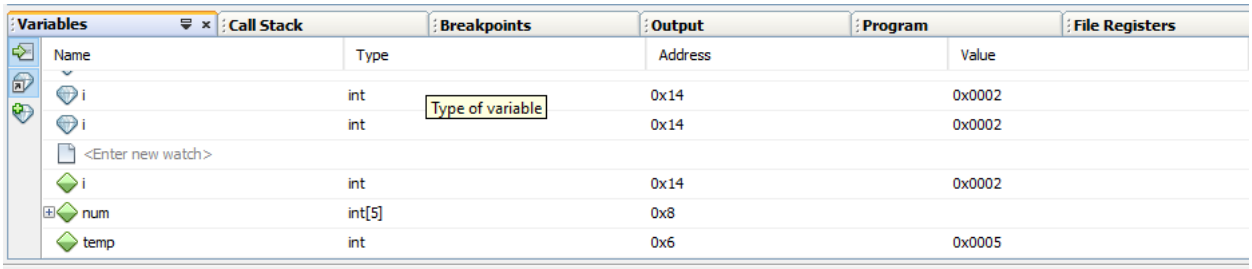



Now we will Debug the code

Debug → Debug Project



We can see the address of num ,i, temp & j variables



**ARMY INSTITUTE OF TECHNOLOGY**
**DEPARTMENT OF INFORMATION TECHNOLOGY**
**(2021-22)**
**LAB PLAN FOR PROGRAMMING SKILL DEVELOPMENT LAB**

**Class:** SE IT
**Subject :** Programming Skill Development Lab
**Academic Year & SEM :** 2021-22 & SEM-2

| Sr. No. | ASSIGNMENT NAME | Submission |
|---|---|---|
| | **Group A** | |
| 1 | Study of Embedded C programming language (Overview, syntax, One simple program likeaddition of two numbers). | 3rd Week of Jan |
| 2 | Write an Embedded C program to add array of n numbers. | 4th Week of January |
| 3 | Write an Embedded C program to transfer elements from one location to another for following:<br>**i)** Internal to internal memory transfer<br>**ii)** Internal to external memory transfer | 1st Week of February |
| | **Group B** | |
| 4 | Write an Embedded C program to interface PIC 18FXXX with LED & blinking it using specifieddelay. | 3rd Week of February |
| 5 | Write an Embedded C program for Timer programming ISR based buzzer on/off. | 1st Week of March |
| 6 | Write an Embedded C program for External interrupt input switch press, output at relay. | 2nd Week of March |
| | **Group C** | |
| 7 | Write an Embedded C program for Generating PWM signal for servo motor/DC motor. | 3rd Week of March |
| 8 | Write an Embedded C program for PC to PC serial communication using UART. | 4th Week of March |
| 9 | Write an Embedded C program for Temperature sensor interfacing using ADC & display on LCD. | 1st Week of April |
| | **Group D** | |
| 10 | Study of Arduino board and understand the OS installation process on Raspberry-pi. | 2nd Week of April |
| 11 | Write simple program using Open source prototype platform like Raspberry-Pi/Beagle board/Arduino for digital read/write using LED and switch Analog read/write using sensor and actuators. | 3rd Week of April |

**Prof. S.D.Samleti**
**(Staff Name )**

**Dr. Sangeeta Jadhav**
**(HOD IT)**

## Expt 6:Write an Embedded C program for interfacing PIC18FXXX to LED and blinking it using specified delay.

**Aim:** To write a C program to interface PIC18F4550 to LED and blink it with a specified delay.

**Experimental Setup:** MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure:**

**Step1:** Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2:** Write the program in C language for interfacing LEDs to PIC18F4550. **(in program properties make sure to add the 0x800 offset)**

**Step3:** Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4:** Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5:** Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6:** Press reset button and execute the program.

**Result:** Check if the LEDs are blinking. You can change the delay and vary the blinking rate.

## Program:

```
#include <p18f4550.h>
void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
        for(j=0;j<5000;j++);
}
void main(void)
{
   TRISB = 0x00;
   LATB = 0xFF;
   while(1)                                    //Loop forever;
   {
       LATB = ~LATB;
       delay(200);
   }
}
```

## Expt 7:Write an Embedded C program for ISR based buzzer on/off using Timer.

**Aim**: To write a C program to interface PIC18F4550 to Buzzer and switch it ON/OFF using Timer ISR..

**Experimental Setup**: MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure**:

**Step1**: Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2**: Write the program in C language for interfacing Buzzer to PIC18F4550, using Timer ISR. **(in program properties make sure to add the 0x800 offset)**

**Step3**: Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4**: Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5**: Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6**: Press reset button and execute the program.

**Result**: Check if the buzzer is sounding ON/OFF and the ISR is getting executed with the specified timer delay. You can change the delay and vary the sounding rate.

## Program:

```
#include <pic18f4550.h>       /* Contains PIC18F4550 specifications */
#define Buzzer LATAbits.LATA5             /* Define buzzer pin */
unsigned int count = 0;

void interrupt Timer1_ISR()
{
    if(TMR1IF==1)
    {
    //1 ms delay time in timer
    TMR1L = 0x20;
    TMR1H = 0xD1;
    count ++;

    if (count >= 1000) //measure upto 1000 ms i.e. 1 seconds
    {
        Buzzer = ~Buzzer;     /* Toggle buzzer pin */
        count = 0;  //reset count
    }
    TMR1IF = 0; //timer1 overflow flag to 0
    }
}


void main()
{
    TRISB=0;                  /* Set as output port */
    TRISAbits.TRISA5 = 0;       //set buzzer pin RA5 as output
    GIE=1;                      /* Enable Global Interrupt */
    PEIE=1;                     /* Enable Peripheral Interrupt */
    TMR1IE=1;                   /* Enable Timer1 Overflow Interrupt */
    TMR1IF=0;

    /* Enable 16-bit TMR1 register,no pre-scale,internal clock, timer OFF */
    T1CON=0x80;         /*1:8 prescale*/
    TMR1L = 0x20;
    TMR1H = 0xD1;
    TMR1ON=1;           /* Turn ON Timer1 */

    while(1);
}
```

## Expt 8:Write an Embedded C program for External Interrupt input switch press, output at Relay.

**Aim**: To write a C program to interface PIC18F4550 to Relay and switch it ON/OFF using input from external switch. Use ISR programming for External Interrupt.

**Experimental Setup:** MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure**:

**Step1**: Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2**: Write the program in C language for interfacing Relay to PIC18F4550, using External Interrupt ISR. **(in program properties make sure to add the 0x800 offset)**

**Step3**: Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4**: Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5**: Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6**: Press reset button and execute the program.

**Result**: Check if the Relay is switching ON/OFF when external interrupt switch is pressed and the ISR is getting executed .

## Program:

```
#include <pic18f4550.h>

#define RELAY_PIN LATAbits.LATA4

void interrupt extint_isr(void)
{
    unsigned int i;
    if(INT1F)
    {
        INT1F = 0;
        INT1IE = 0;
        RELAY_PIN = ~RELAY_PIN;
        for(i=0; i<10000; i++);      //small delay for debouncing
        INT1IE = 1;
    }
}

int main()
{
    ADCON1 = 0x0F;          //set pins as Digital
    TRISAbits.TRISA4 = 0;   //set relay pin RA4 as output
    TRISBbits.TRISB1 = 1;   //Interrupt pin as input
    RELAY_PIN = 1;

    INT1IE  =   1;                      //Enable external interrupt INT1
    INTEDG1 =   0;                      //Interrupt on falling edge
    GIE     =   1;                      // Enable global interrupt

    while(1);
}
```

## Expt 9:Write an Embedded C program for LCD interfacing with PIC18Fxxx.

**Aim**: To write a C program to interface PIC18F4550 to 16x2 Character LCD.

**Experimental Setup**: MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure**:

**Step1**: Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2**: Write the program in C language for interfacing 16x2 LCD to PIC18F4550. **(in program properties make sure to add the 0x800 offset)**

**Step3**: Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4**: Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5**: Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6**: Press reset button and execute the program.

**Result**: Check if the characters are getting printed on the LCD screen .

## Program:

```
#include <p18f4550.h>

#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB


void lcd_delay(unsigned int time)
{
 unsigned int i , j ;

    for(i = 0; i < time; i++)
    {
            for(j=0;j<100;j++);
    }
}


void SendInstruction(unsigned char command)
{
     LCD_RS = 0;         // RS low : Instruction
     LCDPORT = command;
     LCD_EN = 1;         // EN High
     lcd_delay(10);
     LCD_EN = 0;         // EN Low; command sampled at EN falling edge
     lcd_delay(10);
}

void SendData(unsigned char lcddata)
{
     LCD_RS = 1;         // RS HIGH : DATA
     LCDPORT = lcddata;
     LCD_EN = 1;         // EN High
     lcd_delay(10);
     LCD_EN = 0;         // EN Low; data sampled at EN falling edge
     lcd_delay(10);
}

void InitLCD(void)
{
    ADCON1 = 0x0F;
    TRISB = 0x00; //set data port as output
    TRISAbits.RA0 = 0; //RS pin
    TRISAbits.RA1 = 0; // EN pin

    SendInstruction(0x38);      //8 bit mode, 2 line,5x7 dots
    SendInstruction(0x06);    // entry mode
    SendInstruction(0x0C);    //Display ON cursor OFF
    SendInstruction(0x01);      //Clear display
    SendInstruction(0x80);      //set address to 1st line

}
```

```
unsigned char *String1 = " Microembedded";
unsigned char *String2 = " PIC-18F Board";

void main(void)
{
    ADCON1 = 0x0F;
    TRISB = 0x00;       //set data port as output
    TRISAbits.RA0 = 0;  //RS pin
    TRISAbits.RA1 = 0;  // EN pin

    SendInstruction(0x38);      //8 bit mode, 2 line,5x7 dots
    SendInstruction(0x06);      // entry mode
    SendInstruction(0x0C);      //Display ON cursor OFF
    SendInstruction(0x01);      //Clear display
    SendInstruction(0x80);      //set address to 1st line

 while(*String1)
 {
  SendData(*String1);
  String1++;
 }

 SendInstruction(0xC0);      //set address to 2nd line
 while(*String2)
 {
  SendData(*String2);
  String2++;
 }

 while(1);

}
```

## Expt 10: Write an Embedded C program for generating PWM signal for DC/Servo motor on PIC18Fxxx.

**Aim**: To write a C program to interface PIC18F4550 to DC motor and varying speed using PWM signal generation.

**Experimental Setup**: MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure**:

**Step1**: Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2**: Write the program in C language for interfacing DC motor to PIC18F4550 and varying speed using PWM . **(in program properties make sure to add the 0x800 offset)**

**Step3**: Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4**: Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5**: Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6**: Press reset button and execute the program.

**Result**: Check if the DC motor speed varies .

## Program:

```
#include<p18f4550.h>

unsigned char count=0;
bit TIMER,SPEED_UP;

void timer2Init(void)
{
    T2CON   =   0b00000010;             //Prescalar = 16; Timer2 OFF
    PR2     =   0x95;                   //Period Register
}


void delay(unsigned int time)
{
    unsigned int i,j;
    for(i=0;i<time;i++)
        for(j=0;j<1000;j++);
}


void main(void)
{
    unsigned int i;
    TRISCbits.TRISC1    = 0;            //RC1 pin as output
    TRISCbits.TRISC2    = 0;            //CCP1 pin as output
    LATCbits.LATC1      = 0;
    CCP1CON    =     0b00111100;         //Select PWM mode; Duty cycle LSB
    CCP1CON<4:5> = <1:1>
    CCPR1L   =   0x0F;                  //Duty cycle 10%
    timer2Init();                       //Initialise Timer2
    TMR2ON = 1;                         //Timer2 ON

    while(1)                           //Loop forever
    {
        for(i=15;i<150;i++)
        {
            CCPR1L = i;
            delay(100);
        }
        for(i=150;i>15;i--)
        {
            CCPR1L = i;
            delay(100);
        }
    }
}
```

## Expt 11: Write an Embedded C program for PC communication using serial interface (UART).

**Aim**: To write a C program to interface PIC18F4550 to PC using serial communication and transmit / receive characters over it.

**Experimental Setup**: MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure**:

**Step1**: Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2**: Write the program in C language for interfacing PC to PIC18F4550 and sending ascii characters over serial communication. **(in program properties make sure to add the 0x800 offset)**

**Step3**: Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4**: Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5**: Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6**: Press reset button and execute the program.

**Result**: connect the PC to the board using the USB cable. Start a terminal program on the PC (tera Term, Putty, hyperterminal) with the specified baud rate (9600). Check if you are getting the transmitted characters from the board and back .

**Program:**

```
#include<p18F4550.h>
#include<stdio.h>
#define Fosc 48000000UL

void InitUART(unsigned int baudrate)
{
  TRISCbits.RC6 = 0;                        //TX pin set as output
  TRISCbits.RC7 = 1;                        //RX pin set as input
//Non-inverted data; 8-bit baudrate generator
  SPBRG = (unsigned char)(((Fosc /64)/baudrate)-1);
  BAUDCON = 0b00000000 ;
//Asynchronous 8-bit; Transmit enabled; Low speed baudrate select
  TXSTA = 0b00100000;
//Serial port enabled; 8-bit data; single receive enabled
  RCSTA = 0b10010000;                       }

void SendChar(unsigned char data)
{
    while(TXSTAbits.TRMT == 0);      //Wait while transmit register is empty

    TXREG = data;                           //Transmit data
}

void putch(unsigned char data)
{
    SendChar(data);
}

unsigned char GetChar(void)
{
    while(!PIR1bits.RCIF);      //Wait till receive buffer becomes full
    return RCREG;                           //Returned received data
}


void main(void)
{
    InitUART(9600);

    printf("\r\nHello MicroPIC-18F: Enter any Key from Keyboard\r\n");

    while(1)
    {
      printf("%c! ",GetChar());    //Receive character from PC and echo back
    }

 while(1);
}
```

## Expt 12: Write an Embedded C program for interfacing PIC18FXXX to Temperature sensor interfacing using ADC & display on LCD

**Aim:** To write a C program to interface PIC18F4550 to a temperature sensor (LM35) and display the temperature on LCD.

**Experimental Setup:** MicroPIC18F board, USB cable, Power supply adaptor, MPLABx IDE, PICLoader software.

**Procedure:**

**Step1:** Open MPLABX IDE on the PC for program development and create a new project and save it in a new folder.

**Step2:** Write the program in C language for interfacing temperature sensor (LM35) to PIC18F4550 and display result on LCD. **(in program properties make sure to add the 0x800 offset)**

**Step3:** Build the program and create hex file. In case of errors correct program and rebuild to create hex file.

**Step4:** Prepare the experimental setup by connecting the MicroPIC18F board to the PC using USB cable. Power ON the Board. Check for the USBtoSerial COMx allocated by the PC.

**Step5:** Using the PICLoader Software flash the hex file in the PIC18F4550.

**Step6:** Press reset button and execute the program.

**Result:** Check if the temperature values are displayed on the LCD.

## Program:

```
#include <pic18f4550.h>
#include <stdio.h>

#define LCD_EN LATAbits.LA1
#define LCD_RS LATAbits.LA0
#define LCDPORT LATB

unsigned char str[16];

void lcd_delay(unsigned int time)
{
 unsigned int i , j ;

    for(i = 0; i < time; i++)
    {
       for(j=0;j<100;j++);
    }
}


void SendInstruction(unsigned char command)
{
     LCD_RS = 0;         // RS low : Instruction
     LCDPORT = command;
     LCD_EN = 1;         // EN High
     lcd_delay(10);
     LCD_EN = 0;         // EN Low; command sampled at EN falling edge
     lcd_delay(10);
}

void SendData(unsigned char lcddata)
{
     LCD_RS = 1;         // RS HIGH : DATA
     LCDPORT = lcddata;
     LCD_EN = 1;         // EN High
     lcd_delay(10);
     LCD_EN = 0;         // EN Low; data sampled at EN falling edge
     lcd_delay(10);
}

void InitLCD(void)
{
    ADCON1 = 0x0F;
    TRISB = 0x00; //set data port as output
    TRISAbits.RA0 = 0; //RS pin
    TRISAbits.RA1 = 0; // EN pin

    SendInstruction(0x38);      //8 bit mode, 2 line,5x7 dots
    SendInstruction(0x06);    //entry mode
    SendInstruction(0x0C);    //Display ON cursor OFF
    SendInstruction(0x01);      //Clear display
    SendInstruction(0x80);      //set address to 0
}
```

```
void LCD_display(unsigned int row, unsigned int pos, unsigned char *ch)
{
    if(row==1)
        SendInstruction(0x80 | (pos-1));
    else
        SendInstruction(0xC0 | (pos-1));

    while(*ch)
        SendData(*ch++);
}

void ADCInit(void)
{
//ADC channel 7 input
    TRISEbits.RE2 = 1;
//Ref voltages Vdd & Vss; AN0 - AN7 channels Analog
    ADCON1 = 0b00000111;
//Right justified; Acquisition time 4T; Conversion clock Fosc/64
    ADCON2 = 0b10101110;
}

unsigned short Read_Temp(void)
{
    ADCON0 = 0b00011101;      //ADC on; Select channel;
    GODONE = 1;               //Start Conversion

    while(GO_DONE == 1 );     //Wait till A/D conversion is complete
    return ADRES;             //Return ADC result
}

int main(void)
{
    unsigned int temp;
    InitLCD();
    ADCInit();
    LCD_display(1,1,"Temperature:");
    while(1)
    {
        temp = Read_Temp();
        temp = ((temp * 500) / 1023);
        sprintf(str,"%d'C ",temp);
        LCD_display(2,1,str);
        lcd_delay(9000);
    }
    return 0;
}
```