***NAME : Himanshu Dixit***

***ENROLL NO. : B64178***

***BATCH : B10***

***SOFTWARE DEVELOPMENT FUNDAMENTAL LAB-I(15B17CI171) Assignment Sheet (WEEK-8 PHASE-2)***

***Lab B***

**Q 1)** Define a union ‘student’ which can record the roll number, name and age. The name contains up to 33 characters, roll number requires long integer value and age 2-digit integer value. Write a program to compute the size of this union. Also, write the reason for resulting size.

**Solution:**

#include<stdio.h>

union student{

long rollno;

char name[33];

int age;

}s;

int main()

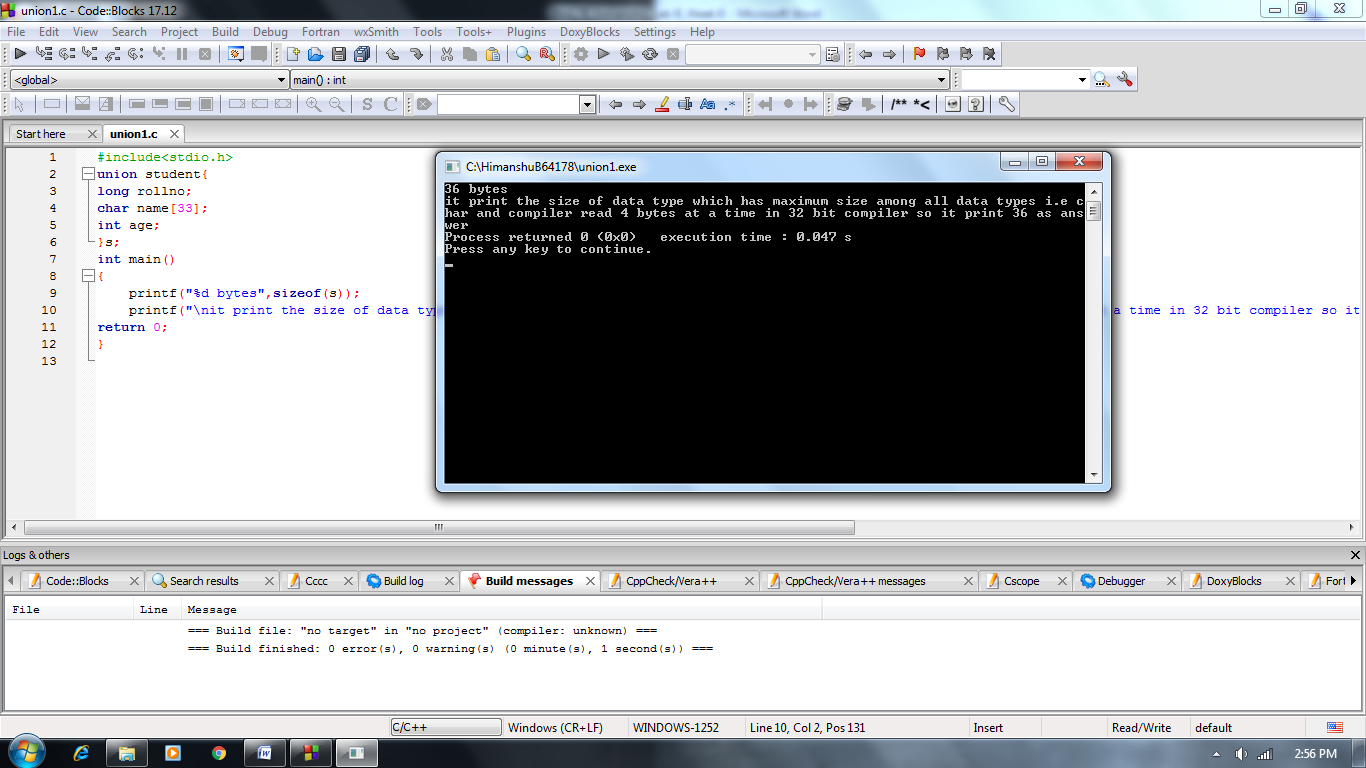
{

printf("%d bytes",sizeof(s));

printf("\nit print the size of data type which has maximum size among all data types i.e char and compiler read 4 bytes at a time in 32 bit compiler so it print 36 as answer");

return 0;

}



**Q 2)** Include courses enrolled in the union given in Q 1. A student can been rolled for maximum 5 courses and 20 characters may be needed to write the full course name. Write a program to compute the size of union. Also, write the reason for resulting size.

**Solution:**

#include<stdio.h>

union student{

long rollno;

char name[33];

int age;

char course[5][20];

}s;

int main()

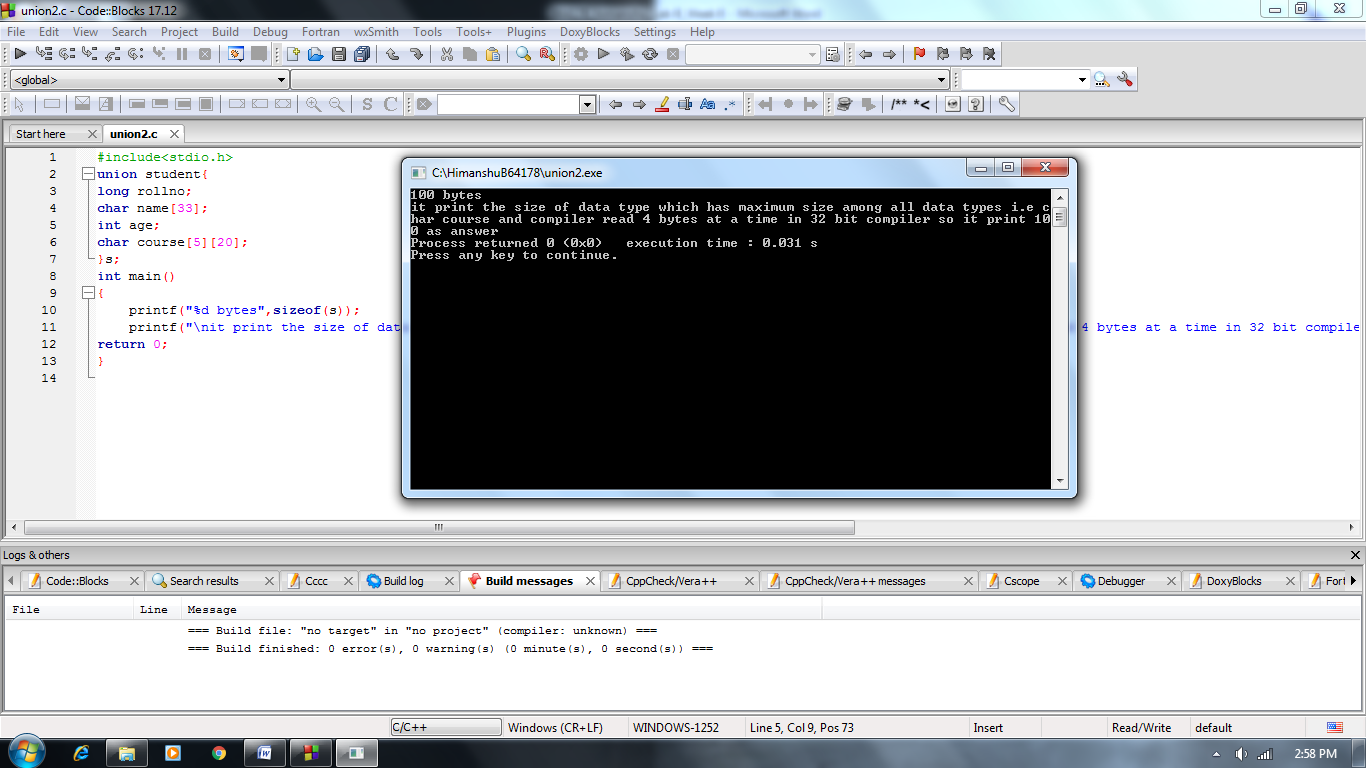
{

printf("%d bytes",sizeof(s));

printf("\nit print the size of data type which has maximum size among all data types i.e char course and compiler read 4 bytes at a time in 32 bit compiler so it print 100 as answer");

return 0;

}



**Q 3)** Use suitable control structure and input the record often students in union defined in Q 2. Print all the records such that no data is lost while printing the output.

**Solution:**

#include<stdio.h>

union student{

long rollno;

char name[33];

int age;

char course[5][20];

}s[10];

int main()

{

for(int i=0;i<3;i++)

{

int noc;

printf("enter student rollnp. ");

scanf("%d",&s[i].rollno);

printf("student rollnp. %d",s[i].rollno);

printf("\nenter student name. ");

scanf("%s",s[i].name);

printf("student name %s",s[i].name);

printf("\nenter student age. ");

scanf("%d",&s[i].age);

printf("student age. %d",s[i].age);

printf("\nenter student no. of course. ");

scanf("%d",&noc);

for(int j=0;j<noc;j++)

{

printf("enter course %d ",j+1);

scanf("%s",s[j].course);

printf("student course %d is %s",j+1,s[j].course);

printf("\n");

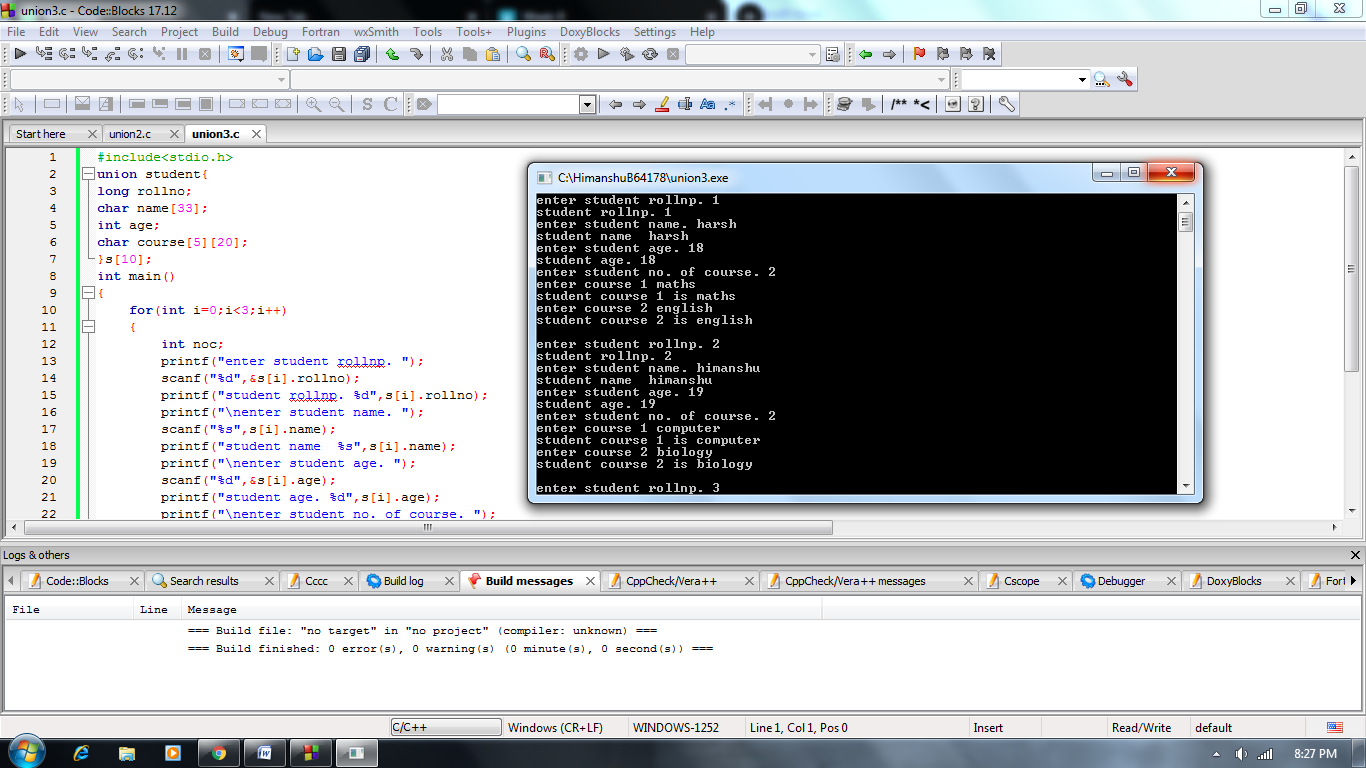
}

printf("\n");

}

return 0;

}



**Q 4)** Define a union ‘numchar’ which can record an integer variable ‘num’ and a character ‘ch’.  Take an integer number from user in num variable. Now, access num and ch values of union using pointer variable. You must use -> symbol to access the values.

**Solution:**

#include <stdio.h>

union numchar {

int num;

char ch;

};

int main(){

union numchar p1;

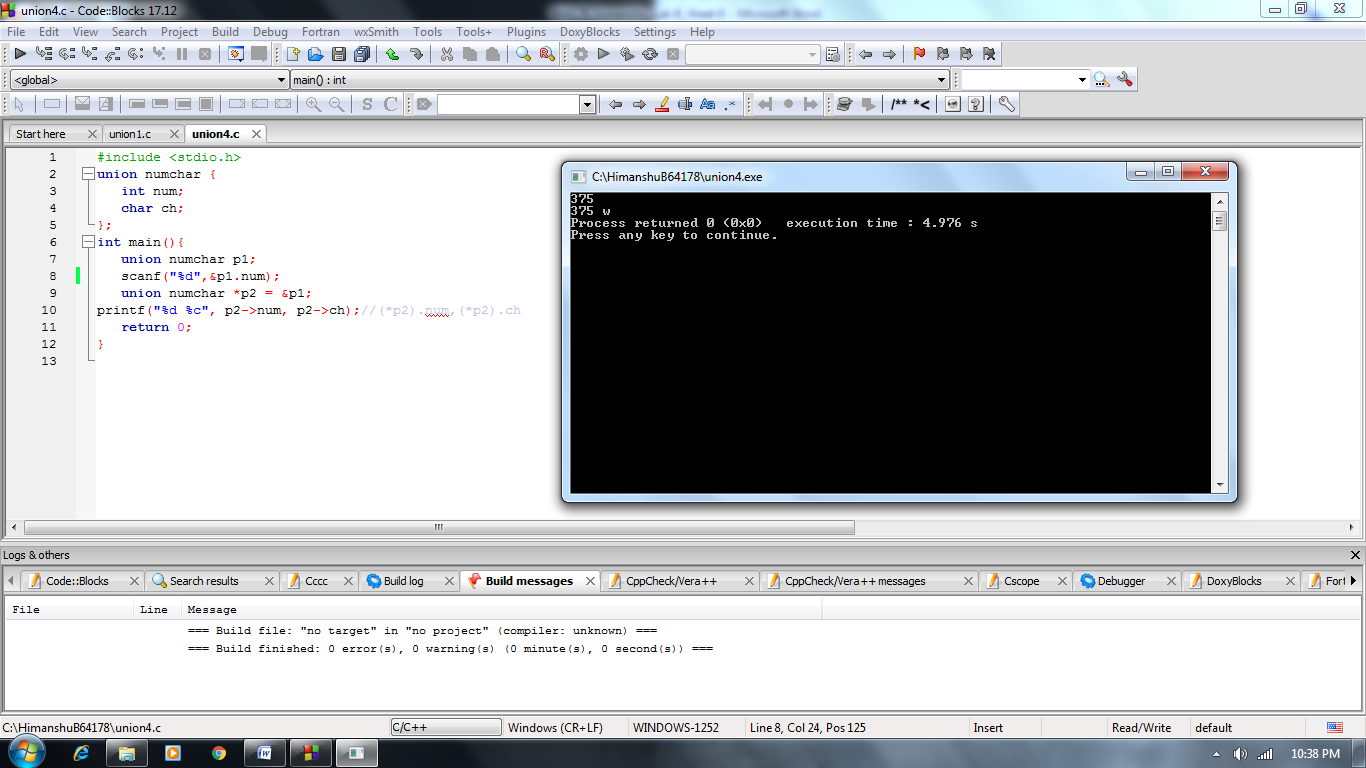
scanf(“%d”,&p1.num);

union numchar \*p2 = &p1;

printf("%d %c", p2->num, p2->ch);

return 0;

}



**Q 5**) Define a union which can record the ID, Name, Age, and Salary of two employees Employee. Using loops, record the input for five users and print all the records such that no data is lost while printing the output.

**Solution:**

#include<stdio.h>

union employee{

int id;

char name[30];

int age;

float salary;

}e[5];

int main()

{

for(int i=0;i<5;i++)

{

printf("enter %d employee details\n",i+1);

printf("enter employee id. ");

scanf("%d",&e[i].id);

printf("employee id. %d",e[i].id);

printf("\nenter employee name. ");

scanf("%s",e[i].name);

printf("employee name %s",e[i].name);

printf("\nenter employee age. ");

scanf("%d",&e[i].age);

printf("employee age. %d",e[i].age);

printf("\nenter employee salary. ");

scanf("%f",&e[i].salary);

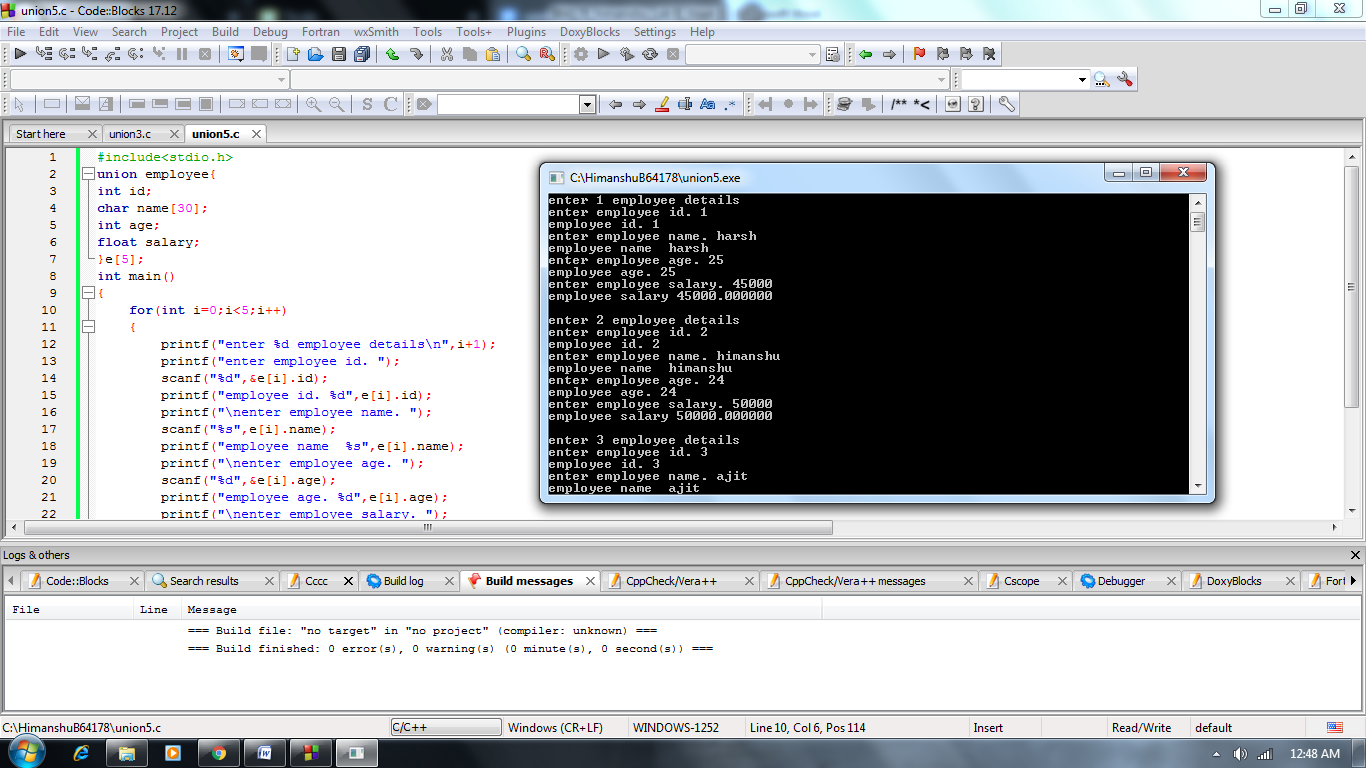
printf("employee salary %f",e[i].salary);

printf("\n\n");

}

return 0;

}

****

**Q 6)** Define a union with three members – one each of char, int, and float data types. The char variable should be able to handle a string of a maximum of 10 characters. Declare two variables of this union.

1. The first variable read user input for all the three members first and then print their values to the output screen.
2. For the second variable, one by one, read input for one member and then print its value.
3. Print the size of the union variables.

Examine the difference in the outputs of the first and second steps.

**Solution:**

#include<stdio.h>

union a

{

char x[10];

int y;

float z;

}t1,t2;

int main()

{

printf("Input values for union members in sequence of string,int,float:");

scanf("%s%d%f",&t1.x,&t1.y,&t1.z);

printf("\nValues of Union Members:");

printf("\nString=%s, Int=%d, Float=%f",t1.x,t1.y,t1.z);

printf("\n\nInput values for string:");

scanf("%s",&t2.x);

printf("\nString Input is: %s",t2.x);

printf("\nInput values for integer:");

scanf("%d",&t2.x);

printf("\nInteger Input is: %d",t2.y);

printf("\nInput values for float:");

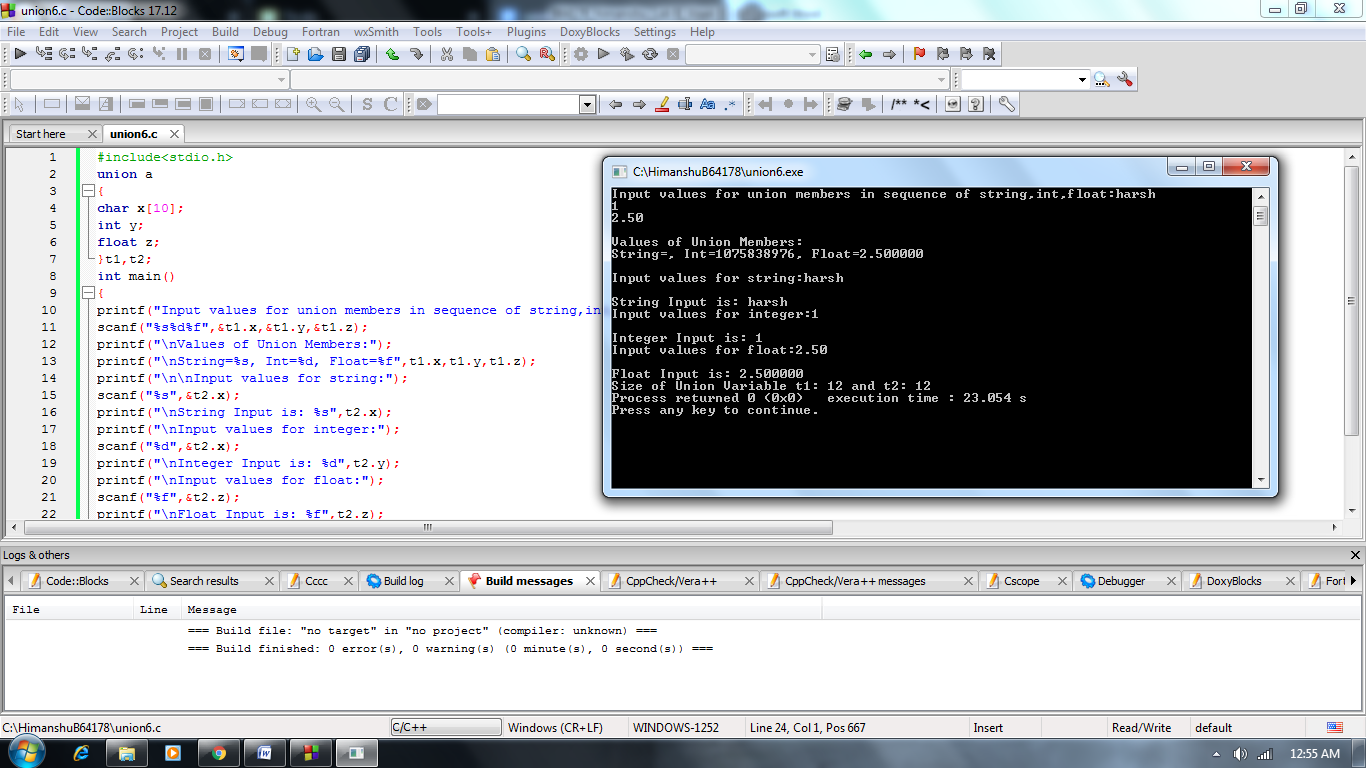
scanf("%f",&t2.z);

printf("\nFloat Input is: %f",t2.z);

printf("\nSize of Union Variable t1: %d and t2: %d",sizeof(t1),sizeof(t2));

return 0;

}



**Q 7)** Write a program in C to demonstrate the difference between union and structure.

1. Define a structure and a union with three members, one each of int, float, and char data-type.
2. Take user input for each member, both for union and structure.
3. Print the value of each member variable and size of structure and union variable.

Examine the difference between the output of structure and union, if any.

**Solution:**

#include <stdio.h>

struct a

{

char x;

int y;

float z;

}m;

union b

{

char p;

int q;

float r;

}n;

int main()

{

printf("Input values for structure members in sequence(char,int,float):");

scanf("%c%d%f",&m.x,&m.y,&m.z);

fflush(stdin);

printf("\nInput values for union members in sequence(char,int,float):");

scanf("%c%d%f",&n.p,&n.q,&n.r);

printf("\nValues of Structure Members:");

printf("\nChar=%c, Int=%d, Float=%f",m.x,m.y,m.z);

printf("\nValues of Union Members:");

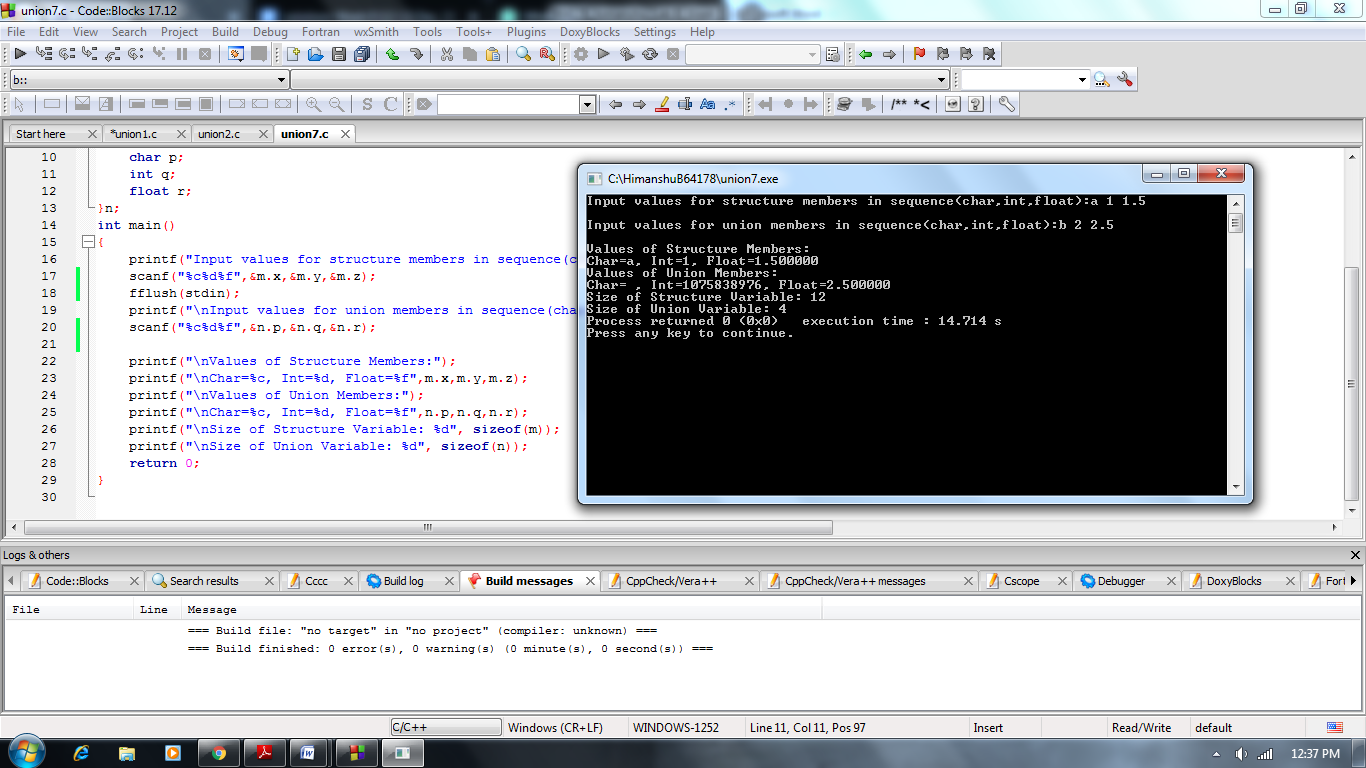
printf("\nChar=%c, Int=%d, Float=%f",n.p,n.q,n.r);

printf("\nSize of Structure Variable: %d", sizeof(m));

printf("\nSize of Union Variable: %d", sizeof(n));

return 0;

}



**Q 8)** In the following C code, we can access the 1st character of the string sval by using \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

   #include <stdio.h>

   struct    {

        char \*name;

      union

       {

           char \*sval;

       } u;

   } symtab[10];

**Solution:**

Symtab[0].u.sval[0]