**Assignment 11 Dec 2017 [Structure & Union ]**

|  |
| --- |
| **Q1**.  #include<stdio.h>  struct arith  {  int sum;  int sub;  int mul;  int div;  int mod;  }result;  int main()  {  int number1,number2;  printf("\n Enter number1 and number2:");  scanf("%d%d",&number1,&number2);  result.sum=sum\_fun(number1,number2);  result.sub=sub\_fun(number1,number2);  result.mul=mul\_fun(number1,number2);  result.div=div\_fun(number1,number2);  result.mod=mod\_fun(number1,number2);  printf("sum=%d\n",result.sum);  printf("sub=%d\n",result.sub);  printf("mul=%d\n",result.mul);  printf("div=%d\n",result.div);  printf("mod=%d\n",result.mod);    return 0;  }  int sum\_fun(int number1,int number2)  {  return number1+number2;  }  int sub\_fun(int number1,int number2)  {  return number1-number2;  }  int mul\_fun(int number1,int number2)  {    return number1\*number2;  }  int div\_fun(int number1,int number2)  {  return number1/number2;  }  int mod\_fun(int number1,int number2)  {  return number1%number2;  } |
| **Output:**  **Enter number1 and number2:5 10**  **sum=15**  **sub=-5**  **mul=50**  **div=0**  **mod=5** |
| **Q2.**  #include<stdio.h>  struct time  {  int hour;  int min;  int sec;  };  struct time start\_time,stop\_time;  int main()  {  int ch;  do  {  printf("\n Enter the start time: Hour:Min:Sec=->");  scanf("%d:%d:%d",&start\_time.hour,&start\_time.min,&start\_time.sec);    if( start\_time.hour==0 && start\_time.min==0 && start\_time.sec==0)  break;  else  {  printf("\n Enter the stop time: Hour:Min:Sec=->");  scanf("%d:%d:%d",&stop\_time.hour,&stop\_time.min,&stop\_time.sec);  printf("\n Difference between time: %d:%d:%d \n",stop\_time.hour-start\_time.hour,stop\_time.min-start\_time.min,stop\_time.sec-start\_time.sec);  }    } while(1);    } |
| **Output:**  **Enter the start time: Hour:Min:Sec=->05:45:35**  **Enter the stop time: Hour:Min:Sec=->08:55:45**  **Difference between time: 3:10:10**  **Enter the start time: Hour:Min:Sec=->0:0:0** |
| **Q3.**  #include<stdio.h>  #include<stdlib.h>  struct complex\_str  {  int real;  int img;  };  struct complex\_str struct1,struct2,sum,sub,mul;  struct complex\_str add()  {  struct complex\_str add\_result;  add\_result.real=struct1.real+struct2.real;  add\_result.img=struct1.img+struct2.img;  return add\_result;  }  struct complex\_str subs()  {  struct complex\_str sub\_result;  sub\_result.real=struct1.real-struct2.real;  sub\_result.img=struct1.img-struct2.img;  return sub\_result;  }  struct complex\_str muls()  {  struct complex\_str mul\_result;    mul\_result.real=(struct1.real\*struct2.real)-(struct1.img\*struct2.img);  mul\_result.img=(struct1.img\*struct2.real)+(struct1.real\*struct2.img);  return mul\_result;  }  int main()  {  int choice,result;  printf("\n Enter first complex number [real] number:");  scanf("%d",&struct1.real);  printf("\n Enter first complex number [Imaginary] number:");  scanf("%d",&struct1.img);  if(struct1.real==0 && struct1.img==0)  exit(0);    printf("\n Enter Second complex number [real] number:");  scanf("%d",&struct2.real);  printf("\n Enter second complex number [Imaginary] number:");  scanf("%d",&struct2.img);  while(1)  {  printf("\n Enter your choice for complex number operation:\n");    printf("\n 1. Add \n 2. Sub \n 3. Mul \n 4. Exit \n => ");  scanf("%d",&choice);  switch(choice)  {    case 1: sum=add();  printf("\n%d+%di",sum.real,sum.img);  break;  case 2: sub=subs();  printf("\n%d-%di",sub.real,sub.img);    case 3:mul=muls();  printf("\n%d+%di",mul.real,mul.img);    case 4:exit(0);  default:printf("\n Please give valid input \n");    }  }  } |
| **Output:**  **Enter first complex number [real] number:4**  **Enter first complex number [Imaginary] number:2**  **Enter Second complex number [real] number:8**  **Enter second complex number [Imaginary] number:7**  **Enter your choice for complex number operation:**  **1. Add**  **2. Sub**  **3. Mul**  **4. Exit**  **=> 1**  **12+9i**  **Enter your choice for complex number operation:**  **1. Add**  **2. Sub**  **3. Mul**  **4. Exit**  **=> 2**  **-4--5i**  **Enter your choice for complex number operation:**  **1. Add**  **2. Sub**  **3. Mul**  **4. Exit**  **=> 3**  **18+44i**  **Enter your choice for complex number operation:**  **1. Add**  **2. Sub**  **3. Mul**  **4. Exit**  **=> 4** |
| **Q4. Write a “C” program to store an integer, string of 4 characters and array of 2 short integers in a union.**  **• Assign integer with Hexadecimal value, access the union with integer, string of characters and array of short integers. And print the addresses and values**  **• Assign a value of Hexadecimal in each character, access the union with integer, string of**  **characters and array of short integers. And print the addresses and values**  **• Assign short integers with Hexadecimal values, access the union with integer, string of**  **characters and array of short integers. And print the addresses and values**  #include<stdio.h>  #include<stdlib.h>  union student  {  int number;  char str[4];  short int number1[2];  };  union student data;  void main()  {  printf("Enter a Hexa decimal number for data.number:");  scanf("%x",&data.number);  printf("number=%d\t&number=%p\nstr=%s\t&str=%p\nnumber1[0]=%hu\t&number1[0]=%p\nnumber1[1]=%hu\t&number1[1]=%p\n",data.number,&data.number,data.str,&data.str,data.number1[0],&data.number1[0],data.number1[1],&data.number1[1]);  printf("Enter a Hexa decimal number for data.str:");  scanf("%x",data.str);  printf("number=%d\t&number=%p\nstr=%s\t&str=%p\nnumber1[0]=%hu\t&number1[0]=%p\nnumber1[1]=%hu\t&number1[1]=%p\n",data.number,&data.number,data.str,&data.str,data.number1[0],&data.number1[0],data.number1[1],&data.number1[1]);  printf("Enter a Hexa decimals number for data.number1[0] and data.number1[1]: ");  scanf("%x%x",&data.number1[0],&data.number1[1]);  printf("number=%d\t&number=%p\nstr=%s\t&str=%p\nnumber1[0]=%hu\t&number1[0]=%p\nnumber1[1]=%hu\t&number1[1]=%p\n",data.number,&data.number,data.str,&data.str,data.number1[0],&data.number1[0],data.number1[1],&data.number1[1]);  } |
| **Output:**  **Enter a Hexa decimal number for data.number:0x3f3f3f**  **number=4144959 &number=0x601044**  **str=??? &str=0x601044**  **number1[0]=16191 &number1[0]=0x601044**  **number1[1]=63 &number1[1]=0x601046**  **Enter a Hexa decimal number for data.str:0x3f3f3f**  **number=4144959 &number=0x601044**  **str=??? &str=0x601044**  **number1[0]=16191 &number1[0]=0x601044**  **number1[1]=63 &number1[1]=0x601046**  **Enter a Hexa decimals number for data.number1[0] and data.number1[1]: 0x3f3f3f**  **3f**  **number=4144959 &number=0x601044**  **str=??? &str=0x601044**  **number1[0]=16191 &number1[0]=0x601044**  **number1[1]=63 &number1[1]=0x601046** |
| **Q5.**  #include<stdio.h>  struct emp  {  int number;  char name[100];  };  int main()  {  struct emp employee={1,"Rajiv"},\*ptr;  ptr=&employee;  printf("\n Employee Number=%d Name=%s \n\n",ptr->number,ptr->name);  return 0;    } |
| **Output:**  **Employee Number=1 Name=Rajiv** |