|  |  |  |
| --- | --- | --- |
| **Name:**  **P K Navin Shrinivas** | **SRN:PES2UG20CS237** | **Section:G-2** |
| **Date:13/05/2021** | **Week Number:2** |

|  |  |
| --- | --- |
| **1** | Write a program to calculate the grade of the student according to the specified marks. Grade A:Marks(>85 and <=100)  Grade B:Marks(>60 and <=85)  Grade C:Marks(>40 and <=60)  Grade D:Marks(>30 and <=40)  Fail: Marks(<30)  **Sample Input:**  Enter your marks:60  **Sample Output:**  You got grade C |
|  | **Program:**  #include<stdio.h>  const char\* grade(int grade)  {  if(grade>85)  return "A";  else if(grade<85 && grade>60)  return "B";  else if(grade<=60 && grade>40)  return "C";  else if(grade<=40 && grade>30)  return "D";  else  return "Fail";  }  int main()  {  printf("Enter grade : ");  int g;  scanf("%i",&g);  printf("Your grade : %s",grade(g));  } |
|  | **Output Screenshot:**  **1.2** |
| **2** | **Write a Program to convert all characters in a given line from lower case to upper case.**  **Sample Input:**  Enter characters to convert case  I am student of 2nd Semester!  **Sample Output**:  I AM STUDENT OF 2ND SEMESTER! |
|  | **Program:**  #include<stdio.h>  int main(){  char str[100000];  printf("Enter a Sentence : \n");  scanf("%[^\n]%c",&str);  int i=0;  while(str[i]!='\0')  {  if(str[i] >= 'a' && str[i] <= 'z')  str[i]=str[i]-32;  i++;  }  printf("Enter string in UPPERCASE: \n %s",str);  } |
|  | **Output Screenshot:**  **2.2** |
| **3** | **Write a C program using bitwise operators for the following:**  i) check whether specified bit is set or not  ii) set the specified bit and print the result  iii) clear the specified bit and print the result  **Sample Input/Output:**  Enter the number which you want check  25  Input number is 25  Enter the bit position, starts from zero  2  bit is not set  Enter the bit position, which you want to set  4  set : 16  The number after set is 25  Enter the bit position, which bit you want to clear  3  set : 0  The number after clear is 17 |
|  | **Program:**  ##include<stdio.h>  #include<stdbool.h>  bool setcheck(int\* num,int\* i)  {  int copy=\*num>>\*i;  if(copy % 2 != 0)  return true;  else  return false;  }  void setbit(int\* num,int\* i)  {  \*num=\*num|1<<\*i;  }  void clearbit(int\* num,int\* i)  {  \*num=\*num&~(1<<\*i);  }  int main()  {  printf("Enter a number :");  int num;  scanf("%i",&num);  int index;  printf("Enter bit whose set condition is to be checked![Start from 0] :");  scanf("%i",&index);  if(setcheck(&num,&index))  printf("The bit is set \n");  else  printf("The bit is not set \n");  printf("Enter bit which is to be set![Start from 0] :");  scanf("%i",&index);  setbit(&num,&index);  printf("Number after setting index %i is : %i \n",index,num);  printf("Enter bit which is to be cleared![Start from 0] :");  scanf("%i",&index);  clearbit(&num,&index);  printf("Number after clearing index %i is : %i \n",index,num);  } |
|  | **Output Screenshot:**  **3.2** |
| **4** | **a)Write a program to generate a multiplication table using for loop**  **b)Write a program to print the following pattern**  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \* |
|  | **Program:**  #include<stdio.h>  int main()  {  for(int i=0;i<5;i++)  {  for(int j=-1;j<i;j++)  printf("\*");  printf("\n");  }  } |
|  | **Output Screenshot:** |
| **5** | **Write a program to implement a Simple Calculator using switch Statement**  **Sample input:**  Enter an operator (+, -, \*,): +  Enter two operands: 3 4  **Sample Output:**  3.0 + 4.0 = 7.0  **Sample input:**  Enter an operator (+, -, \*,): -  Enter two operands: 7 6  **Sample Output:**  7.0 - 6.0 = 1.0 |
|  | **Program:**  #include<stdio.h>  int main(){  char choice;  printf("Select operation [+,-,\*,/] :");  scanf("%c",&choice);  printf("Enter two operands :");  float a,b;  scanf("%f %f",&a,&b);  switch(choice)  {  case '+':  printf("Result : %.2f",a+b);  break;  case '-':  printf("Result : %.2f",a-b);\  break;  case '\*':  printf("Result : %.2f",a\*b);  break;  case '/':  printf("Result : %.2f",a/b);  break;  default:  printf("Please enter a correct choice for operands");  }  } |
|  | **Output Screenshot:**  **5.2** |
| **6** | **Write a program to validate a given date and find the next date**  **Sample input:**  Enter the date 12  Enter the month 12  Enter the year 2000  **Sample Output:**  Date is valid & next date is: 13/12/2000  **Sample input:**  Enter the date 1  Enter the month 13  Enter the year 2000  **Sample Output:**  Month is invalid |
|  | **Program:**  #include<stdio.h>  #include<stdlib.h>  #include<stdbool.h>  bool validdate(int day,int month,int year)  {  if(month>12 || month<=0)  return false;  if(day>31 || day<1)  return false;  if(year%4==0) //corner cases of leap years  {  if(month==2 && day>29)  return false;  else  return true;  }  if(month==2 && day>28) //corner cases of feb month  return false;  if(month<8)  {  if(month%2==0 && day>30)  return false;  }  if(month>=8)  {  if(month%2!=0 && day>30)  return false;  }  return true;  }  int nextday(int day,int month,int year)  {  if(year%4==0) //corner cases of leap years  {  if(month==2 && day==29)  return 1;  }  if(month==2 && day==28) //corner cases of feb month  return 1;  if(month<8)  {  if(month%2==0 && day==30)  return 1;  if(month%2!=0 && day==31)  return 1;  }  if(month>=8)  {  if(month%2!=0 && day==30)  return 1;  if(month%2==0 && day==31)  return 1;  }  return day+1;  }  int nextmonth(int day,int month,int year)  {  if(year%4==0) //corner cases of leap years  {  if(month==2 && day==29)  return month+1;  }  if(month==2 && day==28) //corner cases of feb month  return month+1;  if(month<8)  {  if(month%2==0 && day==30)  return month+1;  }  if(month>=8)  {  if(month%2!=0 && day==30)  return month+1;  if(month==12 && day==31)  return 1;  }  return month;  }  int nextyear(int day,int month,int year)  {  if(month==12 && day==31)  return year+1;  else  return year;  }  int main()  {  char str[10];  int day,month,year;  printf("Enter day : ");  scanf("%i",&day);  printf("Enter month : ");  scanf("%i",&month);  printf("Enter year : ");  scanf("%i",&year);  if(validdate(day,month,year))  {  printf("The given date is valid \n");  int newday=nextday(day,month,year);  int newmonth=nextmonth(day,month,year);  int newyear=nextyear(day,month,year);  printf("Next date : %i/%i/%i",newday,newmonth,newyear);  }  else  printf("The given date is invalid");  } |
|  | **Output Screenshot:**  **6.2** |
| **1** | **Practice Programs**  **Write a program to find the roots of a quadratic equation.**  **Sample Input:**  Enter coefficients a, b and c: 1 2 1  **Sample Output:**  root1 = root2 = -1.00;  **Sample Input:**  Enter coefficients a, b and c: 1 3 1  **Sample Output:**  root1 = -0.38 and root2 = -2.62  **Sample Input:**  Enter coefficients a, b and c: 2 2 1  **Sample Output:**  root1 = -0.50+0.50i and root2 = -0.50-0.50i |
|  | **Program:**  #include<stdio.h>  #include <math.h>  int main(){  int a,b,c;  printf("Enter coefficients a,b,c :");  scanf("%i %i %i",&a,&b,&c);  int d=b\*b-(4\*a\*c);  if(d==0)  {  float root=-(b/(2\*a));  printf("root1=root2=%.3f",root);  }  else if(d>0)  {  float root1=(-b+sqrt(d))/(2\*a);  float root2=(-b-sqrt(d))/(2\*a);  printf("Root 1 =%.3f \n",root1);  printf("Root 2 =%.3f \n",root2);  }  else if(d<0)  {  float real=-b/(2\*a);  float imaginary=sqrt(-d)/(2\*a);//beacause d itself is giving a -ve and sqrt of negtives is well...meh!  printf("root1 =%.3f+i%.3f \n",real,imaginary);  printf("root2 =%.3f-i%.3f \n",real,imaginary);  }  } |
|  | **Output Screenshot:**  **7.2** |
| 2 | **Write a program to squeeze repeated characters by inputting the characters in the given order.**  **Sample Input:**  aaaaabbbbbccccddddeeee  **Sample Output:**  abcde  **Sample Input:**  aaa1111gggg3333  **Sample Output:**  a1g3 |
|  | **Program:**  #include<stdio.h>  #include <string.h>  int main(){  char str[100000];  printf("Input:");  scanf("%s",&str);  char init=str[0];  printf("%c",str[0]);  for(int i=0;i<strlen(str);i++)  {  if(str[i]==init)  continue;  else{  init=str[i];  printf("%c",str[i]);  }  }  } |
|  | **Output Screenshot:**  **8.2** |