|  |  |
| --- | --- |
| *1 . Write a function that will calculate and display the real roots of the quadratic equation*  ***.***  //1803117  #include<stdio.h>  #include<math.h>  void root(void);  int main()  {  root();  return 0;  }  void root(void)  {  float a,b,c;  float x1,x2;  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  printf("Enter the value of b more grater :\n");  scanf("%f %f %f",&a,&b,&c);  }  x1=(-b+sqrt((b\*b)-(4\*a\*c)))/(2\*a);  x2=(-b-sqrt((b\*b)-(4\*a\*c)))/(2\*a);  printf("root1 =%.2f\nroot2 =%.2f\n",x1,x2);  return;  } | *2. Write a program that will calculate the real roots of the quadratic equation*  ***.***  *Read the value of a,b,c in the main portion of the program . Then access the function written for the preceding problem in order to obtain the desired solution .Finally display the coefficient ,followed by the calculated values of x1,x2 .*    //1803117  #include<stdio.h>  #include<math.h>  void root(float a,float b,float c);  int main()  {  float a,b,c;  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  printf("Enter the value of b more grater :\n");  scanf("%f %f %f",&a,&b,&c);  root(a,b,c);  }  else  {  root(a,b,c);  }  return 0;  }  void root(float a,float b,float c)  {  float x1,x2;  x1=(-b+sqrt((b\*b)-(4\*a\*c)))/(2\*a);  x2=(-b-sqrt((b\*b)-(4\*a\*c)))/(2\*a);  printf("a= %f\nb= %f\nc= %f\n",a,b,c);  printf("root1 =%f\nroot2 =%f\n",x1,x2);  return;  } |
| *3. Modify the previous program that can execute all roots of the quadratic equation*  *(complex root also)*  ***.***  //1803117  #include<stdio.h>  #include<math.h>  void rroot(float a,float b,float c);  void uroot(float a,float b,float c);  int main()  {  float a,b,c;  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  uroot(a,b,c);  }  else  {  rroot(a,b,c);  }  return 0;  }  void rroot(float a,float b,float c)  {  float x1,x2;  x1=(-b+sqrt((b\*b)-(4\*a\*c)))/(2\*a);  x2=(-b-sqrt((b\*b)-(4\*a\*c)))/(2\*a);  printf("a= %f\nb= %f\nc= %f\n",a,b,c);  printf("x1 =%f\nx2 =%f\n",x1,x2);  return;  }  void uroot(float a,float b,float c)  {  float x,y1,y2;  x=(-b)/(2\*a);  y1=+(sqrt(4\*a\*c-b\*b))/(2\*a);  y2=-(sqrt(4\*a\*c-b\*b))/(2\*a);  printf("a= %f\nb= %f\nc= %f\n",a,b,c);  printf("x1= (%f)+(%fi)\nx22= (%f)+(%fi)\n",x,y1,x,y2);  return;  } | *5. Write a function that will allow a floating number to be raised to an integer power . In other words, we wish to evaluate the formula* ***.*** *Where y and x are floating – point variables and n is an integer variable .*  //1803117  #include<stdio.h>  #include<math.h>  void cal(void);  int main()  {  cal();  return 0;  }  void cal(void)  {  float y,x;  int n;  printf("Enter the value of x :\n");  scanf("%f",&x);  printf("Enter the value of n :\n");  scanf("%d",&n);  y=pow(x,n);  printf("\n");  printf("\n (%.2f)^(%d)=(%f)\n",x,n,y);  } | | |
| *4.Modify the previous problem that can execute all roots of quadratic equation*  ***.***  *Continuously without stopping by user .*  //1803117  #include<stdio.h>  #include<math.h>  #include<conio.h>  void rroot(float a,float b,float c);  void uroot(float a,float b,float c);  char f;  int main()  {  float a,b,c;  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  uroot(a,b,c);  }  else  {  rroot(a,b,c);  }  return 0;  }  void rroot(float a,float b,float c)  {  float x1,x2;  x1=(-b+sqrt((b\*b)-(4\*a\*c)))/(2\*a);  x2=(-b-sqrt((b\*b)-(4\*a\*c)))/(2\*a);  printf("a= %f\nb= %f\nc= %f\n",a,b,c);  printf("x1 =%f\nx2 =%f\n",x1,x2);  printf("If you continue press 1.....");  f=getche();  printf("\n"); | if(f=='1')  {  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  uroot(a,b,c);  }  else  {  rroot(a,b,c);  }  }  else{printf(" Good bye !\n");}  return;  }  void uroot(float a,float b,float c)  {  float x,y1,y2;  x=(-b)/(2\*a);  y1=+(sqrt(4\*a\*c-b\*b))/(2\*a);  y2=-(sqrt(4\*a\*c-b\*b))/(2\*a);  printf("a= %f\nb= %f\nc= %f\n",a,b,c);  printf("x1= (%f)+(%fi)\nx22= (%f)+(%fi)\n",x,y1,x,y2);  printf("If you continue press 1 .........");  f=getche();  printf("\n");  if(f=='1')  {  printf("Enter a ,b , c :\n");  scanf("%f %f %f",&a,&b,&c);  if(b\*b<4\*a\*c)  {  uroot(a,b,c);  }  else  {  rroot(a,b,c);  }  }  else{printf(" Good bye !\n");}  return;  } | |
| *6.Modify the program (5) that reads the value of x and n without stopping by user and evaluate*  //1803117  #include<stdio.h>  #include<math.h>  #include<conio.h>  double cal(float x,int n);  int main()  { float y, x;double a;  int n;char ch;  while(1){  printf("Enter the value of x :\n");  scanf("%f",&x);  printf("Enter the value of n :\n");  scanf("%d",&n);  a=cal(x,n);  printf("\n");  printf("x= %.2f , n=%d ,(%.2f)^(%d)=(%lf)\n",x,n,x,n,a);  printf("If you continue press 1...");  ch=getche();  printf("\n");  if(ch!='1')  {  break;  }}  printf("\n Good bye !");  return 0;  }  double cal(float x,int n)  {  return (pow(x,n));  } | *7.Write a program that evaluate the equation*    *Read the value of n and x .*  *Print the value of y using function .*  //1803117  #include<stdio.h>  #include<math.h>  #include<conio.h>  double cal(float x,int n);  int main()  { float y, x;double a;  int n;char ch;  while(1){  printf("Enter the value of x :\n");  scanf("%f",&x);  printf("Enter the value of n :\n");  scanf("%d",&n);  a=cal(x,n);  printf("\n");  printf("x= %.2f , n=%d ,e^((%dln(%.2f))=(%lf)\n",x,n,n,x,a);  printf("If you continue press 1...");  ch=getche();  printf("\n");  if(ch!='1')  {  break;  }}  printf("\n Good bye !");  return 0;  }  double cal(float x,int n)  {  return (pow(2.718281828,n\*log(x)));  } | | |

|  |  |
| --- | --- |
| *8. Write a program that determine the roots of an algebraic equation*  *using an iterative procedure using function .*  //1803117  #include<stdio.h>  #include<math.h>  void iter(float guess);  #define true 1  #define false 0  int main()  {  float guess;  printf("Initial guess :\n");  scanf("%f",&guess);  iter(guess);  return 0;  }  void iter(float guess)  {  int flag=true,count=0;  float root,test,error;  while(flag)  {  ++count;  if(count==50)  flag=false;  test=10.0-(3.0\*guess\*guess);  if(test>0){  root=pow(test,0.2);  printf("\nIteration number : %2d",count);  printf(" x=%7.5f",root);  error=fabs(root-guess);  if(error>0.00001)guess=root;  else{  flag=false;  printf("\n\nRoot= %7.5f",root);  printf(" No. of iterations = %2d",count);  }  }  else{  flag=false;  printf("\nNumber out of range - try another initial guess");  }  }  if(count==50 && error>0.00001)  printf("\n\nConvergence not obtained after 50 iterations"); } | *9.Write a program that reads n numbers by a function and return sum .Then calculate the average of n numbers .*  //1803117  #include<stdio.h>  float summe(int n);  int main()  {int n;  float avg;  printf("Enter the number of total value :\n");  scanf("%d",&n);  avg=summe(n)/n;  printf("Average = %.3f\n",avg);  return 0;  }  float summe(int n)  {int i;  float sum=0,a;  printf("Enter %d number :\n",n);  for(i=1;i<=n;i++)  {scanf("%f",&a);  sum=sum+a; }  return sum; }  *10 . Write a program that execute the compound interest using function .That the values of p,r and n are entered as arguments and the calculated value of F is returned .*  //1803117  #include<stdio.h>  #include<math.h>  float cominter(float p,float r,float n);  int main()  {  float p,r,n;  printf("Enter the value of p :\n");  scanf("%f",&p);  printf("Enter the value of r :\n");  scanf("%f",&r);  printf("Enter the value of n :\n");  scanf("%f",&n);  printf("Compound Interest = %f\n",cominter(p,r,n));  return 0;  }  float cominter(float p,float r,float n)  {  float f;  f=p\*pow((1+(r/100)),n);  return f; } |