|  |  |
| --- | --- |
| 1(a).Write a program that illustrates the relationship between two floating variables(where u = 0.3) ,their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  int main()  {  float u=0.3;  float v;  float \*pu;  float \*pv;  pu=&u;  v=\*pu;  pv=&v;  printf("\nu=%f &u=%x pu=%x \*pu=%f",u,&u,pu,\*pu);  printf("\n\nv=%f &v=%x pv=%x \*pv=%f",v,&v,pv,\*pv);  return 0;  } | 1.(b).Write a program that illustrates the relationship between two double precision variables (where u = 0.3 x ),their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  #include<math.h>  int main()  {  double u=0.3\*pow(10,45);  double v;  double \*pu;  double \*pv;  pu=&u;  v=\*pu;  pv=&v;  printf("\nu=%lf &u=%x pu=%x \*pu=%lf",u,&u,pu,\*pu);  printf("\n\nv=%lf &v=%x pv=%x \*pv=%lf",v,&v,pv,\*pv);  return 0;  } |

|  |  |
| --- | --- |
| 1.(c).Write a program that illustrates the relationship between two character variables(where u= ‘C’) ,their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  int main()  {  char u='C';  char v;  char \*pu;  char \*pv;  pu=&u;  v=\*pu;  pv=&v;  printf("\nu=%c &u=%x pu=%x \*pu=%c",u,&u,pu,\*pu);  printf("\n\nv=%c &v=%x pv=%x \*pv=%c",v,&v,pv,\*pv);  return 0;  } | 2 (a). Write a program that illustrates the relationship between two floating variables(where v = 0.3) ,their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  int main()  {  float v=0.3;  float \*pv;  pv=&v;  printf("\n\*pv=%f v=%f",\*pv,v);  \*pv=0;  printf("\n\n\*pv=%f v=%f",\*pv,v);  return 0;  } |

|  |  |  |
| --- | --- | --- |
| 2.(b) . Write a program that illustrates the relationship between two double precision variables (where v = 0.3 x ),their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  #include<math.h>  int main()  {  double v=0.3\*pow(10,45);  double \*pv;  pv=&v;  printf("\n\*pv=%lf v=%lf",\*pv,v);  \*pv=0;  printf("\n\n\*pv=%lf v=%lf",\*pv,v);  return 0; } | 2.(c) Write a program that illustrates the relationship between two character variables(where v= ‘C’) ,their corresponding addresses and their associated pointers .  //1803117  #include<stdio.h>  int main()  {  char v='C';  char \*pv;  pv=&v;  printf("\n\*pv=%c v=%c",\*pv,v);  \*pv='A';  printf("\n\n\*pv=%c v=%c",\*pv,v);  return 0;  } | |
| 3. Write a program where a single one – dimensional, character – type array is passed to func1 .Initially, assign the string “red” to the array within main . The reassign the string “green” in func1 . Then print this string in main before and after calling func1() , and print in func1 .  //1803117  #include<stdio.h>  void func1(char \*a[]);  int main()  {  char \*a[]={"red"};  printf("\nBefore calling finc1 : %s",\*a);  func1(a);  printf("\nAfter calling func1 : %s",\*a);  return 0;  }  void func1(char \*a[])  {  \*a="green";  printf("\nWithin func1 : %s",\*a);  return;  } | 4. Write a program that counts the number of words and the total number of characters in the line of text ,using pointer .  //1803117  #include<stdio.h>  #include<string.h>  void scan\_line(char l[],int \*w,int \*c);  int main()  {  char l[100];  int words=0;  int c=0;  printf("Enter the line of text :\n");  gets(l);  printf("\n");  scan\_line(l,&words,&c);  printf("Number of words : %d\n",words);  printf("\n");  printf("Number of characters : %d\n",c);  printf("\n");  return 0;  }  void scan\_line(char l[],int \*w,int \*c)  { int i=0;  while(1)  {  if(l[i]==' '|| l[i]=='\0'){  ++(\*w);  }  if((l[i]>='A' && l[i]<='Z') ||(l[i]>='a' && l[i]<='z')) {  ++(\*c);  }  if(l[i]=='\0'){  break; }  i++; }  return; } | |
| 5.(a) Write a program that can process multiple line of text .Then determine the number of vowels,consonants,digits,whitespace characters and ‘’other’’ characters for each line .Finally , determine the average number of vowels and consonants per line .Using 2D character array to store line of text and pointer .  //1803117  #include<stdio.h>  #include<ctype.h>  #include<string.h>  void scan\_line(char line[1000],int \*pv,int \*pc,int \*pd,int \*pw,int \*po);  int main() {  char line[100][1000];  int v,c,d,w,o,i=0;  for(i=0;;i++)  { v=0;c=0;d=0;w=0;o=0;  printf("\nEnter a line of text if u want stopping press 'End' :\n"); gets(line[i]);  if(strcmp(line[i],"End")==0){  break; }  else if(strcmp(line[i],"End")!=0) {  scan\_line(line[i],&v,&c,&d,&w,&o);  printf("\nNo of vowels : %d",v);  printf("\nNo of consonants : %d",c);  printf("\nNo of digits : %d",d);  printf("\nNo of whitespace : %d",w);  printf("\nNo of other character : %d",o);  printf("\nAverage vowels : %f",(float)(v+c)/v); printf("\nAverage consonants : %f",(float)(v+c)/c);} }  return 0; }  void scan\_line(char line[1000],int \*pv,int \*pc,int \*pd,int \*pw,int \*po)  {char c; int i; i=0;  while(line[i]!='\0')  {c=toupper(line[i]);  if(c=='A'||c=='E'||c=='I'||c=='O'||c=='U'){  ++(\*pv); }else if(c>='A' && c<='Z'){++(\*pc); } else if(c>='0' && c<='9') {  ++(\*pd);}else if(c==' '||c=='\t'){++(\*pw);}  else{++(\*po); }  i++; } return; } | 5.(b) Write a program that can process multiple line of text . Then determine the number of vowels,consonants,digits,whitespace characters and ‘’other’’ characters for each line .Finally , determine the average number of vowels and consonants per line .Store the multiple line of text as individual strings and maintain a pointer to each string within a one – dimensional array of pointer .  //1803117  #include<stdio.h>  #include<ctype.h>  #include<string.h>  void scan\_line(char \*line,int \*pv,int \*pc,int \*pd,int \*pw,int \*po);  int main()  {char line[1000]; int v,c,d,w,o,i=0;  for(i=0;;i++)  { v=0;c=0;d=0;w=0;o=0;  printf("\nEnter a line of text if u want stopping press 'End' :\n");  gets(line);  if(strcmp(line,"End")==0) { break;}  else if(strcmp(line,"End")!=0) {  scan\_line(line,&v,&c,&d,&w,&o);  printf("\nNo of vowels : %d",v);  printf("\nNo of consonants : %d",c);  printf("\nNo of digits : %d",d);  printf("\nNo of whitespace : %d",w);  printf("\nNo of other character : %d",o);  printf("\nAverage vowels : %f",(float)(v+c)/v);  printf("\nAverage consonants : %f",(float)(v+c)/c);  line[1000]='0'; } } return 0; }  void scan\_line(char \*line,int \*pv,int \*pc,int \*pd,int \*pw,int \*po)  {char c;  int i;  i=0;  while(line[i]!='\0')  {  line[i]=toupper(line[i]);  if(line[i]=='A'||line[i]=='E'||line[i]=='I'||line[i]=='O'||line[i]=='U')  { ++(\*pv); }  else if(line[i]>='A' && line[i]<='Z') {  ++(\*pc); }  else if(line[i]>='0' && line[i]<='9') {  ++(\*pd); }  else if(line[i]==' '||line[i]=='\t') {  ++(\*pw); }  else {  ++(\*po); }  i++; }  return; } | |
| 6.Write a program that illustrates the relation between array elements and their addresses . Use static long integer array to store elements .  //1803117  #include<stdio.h>  int main()  { static long int x[10]={11111111,22222222,  333333333,444444444,  555555555,666666666,  77777777,88888888,  99999999,10000000};  int i;  for(i=0;i<=9;i++)  {  printf("\ni=%d x[i]=%ld \*(x+i)=%ld",i,x[i],\*(x+i));  printf(" &x[i]=%x x+i=%x",&x[i],(x+i));  }  return 0;  } | 7.(a)Write a program that reorder integer from smallest to largest by magnitude .Use one dimensional pointer integer array .  //1803117  #include<stdio.h>  #include<stdlib.h>  void reorder(int n,int \*a);  int main()  {  int i,n,\*a,t;  printf("\nHow many number will entered ?\n");  scanf("%d",&n);  a=(int \*)malloc(n\*sizeof(int));  printf("Enter the elements :\n");  for(i=0;i<n;i++)  {  scanf("%d",a+i);  \*(a+i)=abs(\*(a+i));  }  reorder(n,a);  printf("\nReordered list :\n");  for(i=0;i<n;i++)  {  printf("i=%d a=%d\n",i+1,\*(a+i));  }  return 0;  }  void reorder(int n,int \*a)  {  int i,temp,j;  for(i=0;i<n;i++)  for(j=i+1;j<n;j++)  {  if(\*(a+i)>\*(a+j))  {  temp=\*(a+i);  \*(a+i)=\*(a+j);  \*(a+j)=temp;  }  }  return;  } |

|  |  |
| --- | --- |
| 7.(b) Write a program that reorder integer from smallest to largest by algebraic .Use one dimensional pointer integer array .  //1803117  #include<stdio.h>  #include<stdlib.h>  void reorder(int n,int \*a);  int main()  {  int i,n,\*a,t;  printf("\nHow many number will entered ?\n");  scanf("%d",&n);  a=(int \*)malloc(n\*sizeof(int));  printf("Enter the elements :\n");  for(i=0;i<n;i++)  {  scanf("%d",a+i);  }  reorder(n,a);  printf("\nReordered list :\n");  for(i=0;i<n;i++)  {  printf("i=%d a=%d\n",i+1,\*(a+i));  }  return 0;  }  void reorder(int n,int \*a)  {  int i,temp,j;  for(i=0;i<n;i++){  for(j=i+1;j<n;j++)  {  if(\*(a+i)>\*(a+j))  {  temp=\*(a+i);  \*(a+i)=\*(a+j);  \*(a+j)=temp;  }  }  }  return;  } | 7(c) Write a program that reorder integer from largest to smallest by magnitude .Use one dimensional pointer integer array .  //1803117  #include<stdio.h>  #include<stdlib.h>  void reorder(int n,int \*a);  int main()  {  int i,n,\*a,t;  printf("\nHow many number will entered ?\n");  scanf("%d",&n);  a=(int \*)malloc(n\*sizeof(int));  printf("Enter the elements :\n");  for(i=0;i<n;i++)  {  scanf("%d",a+i);  \*(a+i)=abs(\*(a+i));  }  reorder(n,a);  printf("\nReordered list :\n");  for(i=0;i<n;i++)  {  printf("i=%d a=%d\n",i+1,\*(a+i));  }  return 0;  }  void reorder(int n,int \*a)  {  int i,temp,j;  for(i=0;i<n;i++){  for(j=i+1;j<n;j++)  {  if(\*(a+i)<\*(a+j))  {  temp=\*(a+i);  \*(a+i)=\*(a+j);  \*(a+j)=temp;  }  }  }  return;  } |

|  |  |
| --- | --- |
| 7(d) ) Write a program that reorder integer from largest to smallest by algebric .Use one dimensional pointer integer array .  //1803117  #include<stdio.h>  #include<stdlib.h>  void reorder(int n,int \*a);  int main()  {  int i,n,\*a,t;  printf("\nHow many number will entered ?\n");  scanf("%d",&n);  a=(int \*)malloc(n\*sizeof(int));  printf("Enter the elements :\n");  for(i=0; i<n; i++)  {  scanf("%d",a+i);  }  reorder(n,a);  printf("\nReordered list :\n");  for(i=0; i<n; i++)  {  printf("i=%d a=%d\n",i+1,\*(a+i));  }  return 0;  }  void reorder(int n,int \*a)  {  int i,temp,j;  for(i=0; i<n; i++){  for(j=i+1; j<n; j++)  {  if(\*(a+i)<\*(a+j))  {  temp=\*(a+i);  \*(a+i)=\*(a+j);  \*(a+j)=temp;  }  }  }  return;  } |  |

|  |
| --- |
| 8. Write a program that add two tables of numbers and calculated each element is larger of the corresponding elements in other two tables .Use two dimensional pointer array to store and calculation.  //1803117  #include<stdio.h>  #include<stdlib.h>  #define maxrows 20  void read(float \*a[maxrows],int m,int n);  void compute(float \*a[maxrows],float \*b[maxrows],float \*c[maxrows],int m,int n);  void out(float \*c[maxrows],int m,int n);  int i,j;  int main(){  float \*a[maxrows],\*b[maxrows],\*c[maxrows];  int rows,cols;  printf("Enter the rows number :\n");  scanf("%d",&rows);  printf("Enter the colms number :\n");  scanf("%d",&cols);  for(i=0;i<rows;i++){  a[i]=(float \*)malloc(cols\*sizeof(float));  b[i]=(float \*)malloc(cols\*sizeof(float));  c[i]=(float \*)malloc(cols\*sizeof(float));}  printf("Enter the elements of A table :\n");  read(a,rows,cols);  printf("Enter the elements of B table :\n");  read(b,rows,cols);  compute(a,b,c,rows,cols);  printf("Show table C:\n");  out(c,rows,cols);  return 0; }  void read(float \*a[maxrows],int m,int n){  for(i=0;i<m;i++)  {printf("Enter row %d\n",i+1);  for(j=0;j<n;j++) {  scanf("%f",(\*(a+i)+j));  \*(\*(a+i)+j)=abs(\*(\*(a+i)+j)); } } return; }  void compute(float \*a[maxrows],float \*b[maxrows],float \*c[maxrows],int m,int n) {  for(i=0;i<m;i++) {  for(j=0;j<n;j++) {  \*(\*(c+i)+j)=\*(\*(a+i)+j)+\*(\*(b+i)+j); } } return; }  void out(float \*c[maxrows],int m,int n) {  for(i=0;i<m;i++) {  for(j=0;j<n;j++) {  printf(" %.3f",\*(\*(c+i)+j)); }  printf("\n");  }  } |

|  |
| --- |
| 9. Write a program that add two tables of numbers and calculated each element is larger of the corresponding elements in other two tables .Use one dimensional pointer array to store and calculation.  //1803117  #include<stdio.h>  #include<stdlib.h>  #define maxrows 50  void read(float \*a[maxrows],int m,int n);  void compute(float \*a[maxrows],float \*b[maxrows],float \*c[maxrows],int m,int n);  void out(float \*c[maxrows],int m,int n);  int i,j;  int main() { float \*a[maxrows],\*b[maxrows],\*c[maxrows];  int rows,cols;  printf("Enter the rows number :\n");  scanf("%d",&rows);  printf("Enter the colms number :\n");  scanf("%d",&cols);  for(i=0;i<rows;i++){  a[i]=(float \*)malloc(cols\*sizeof(float));  b[i]=(float \*)malloc(cols\*sizeof(float));  c[i]=(float \*)malloc(cols\*sizeof(float)); }  printf("Enter the elements of A table :\n");  read(a,rows,cols);  printf("Enter the elements of B table :\n");  read(b,rows,cols);  compute(a,b,c,rows,cols);  printf("Show table C:\n");  out(c,rows,cols);  return 0; }  void read(float \*a[maxrows],int m,int n)  { for(i=0;i<m;i++)  {printf("Enter row %d\n",i+1);  for(j=0;j<n;j++) {  scanf("%f",(a[i]+j));  \*(a[i]+j)=abs(\*(a[i]+j)); } } return; }  void compute(float \*a[maxrows],float \*b[maxrows],float \*c[maxrows],int m,int n)  {  for(i=0;i<m;i++) {  for(j=0;j<n;j++) {  \*(c[i]+j)=\*(a[i]+j)+ \*(b[i]+j); } }  return;  }  void out(float \*c[maxrows],int m,int n)  { for(i=0;i<m;i++) {  for(j=0;j<n;j++)  { printf(" %.3f",\*(c[i]+j));  }  printf("\n"); }  return; } |

|  |  |
| --- | --- |
| 10.Write a program that sort a list of strings into alphabetical order or reverse-alphabetical order select by user any one of this .  //1803117  #include<stdio.h>  #include<stdlib.h>  #include<string.h>  #include<conio.h>  int main()  {  int i,j,n;  printf("Enter the total number of string :\n");  scanf("%d",&n);  printf("\n");  char \*a[n],\*temp,ch;  printf("Enter %d string :\n",n);  for(i=0; i<=n; i++)  {  a[i]=(char \*)malloc(100 \* sizeof(char));  gets(a[i]);  }  printf("\n");  printf("Enter 'a' to arrange alphabetical or 'r' to reverse-alphabetical:....");  ch=getche();  if(ch=='a')  {  for(i=0; i<=n; i++)  {  for(j=i+1; j<=n; j++)  {  if(strcmp(a[i],a[j])>0)  {  temp=a[i];  a[i]=a[j];  a[j]=temp;  }  else  continue;  }  }  printf("\nLower to upper string :\n");  printf("\n");  for(i=0; i<=n; i++)  {  printf("%s\n",a[i]);  } | }  else if(ch=='r')  {  for(i=0; i<=n; i++)  {  for(j=i+1; j<=n; j++)  {  if(strcmp(a[i],a[j])<0)  {  temp=a[i];  a[i]=a[j];  a[j]=temp;  }  else  continue;  }  }  printf("\nUpper to lower string :\n");  printf("\n");  for(i=0; i<=n; i++)  {  printf("%s\n",a[i]);  }  }  return 0;  } |