1. W.A.P. to implement string concatenation using strcat().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Prasad " ,str2[]="is Embedded Developer";

strcat(str1,str2);

printf("%s",str1);

}

1. W.A.P. to implement string using strcpy().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Prasad",str2[10];

strcpy(str2,str1);

printf("%s",str2);

}

1. W.A.P. to implement string using strncpy().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Prasad is Embedded Engineer";

char str2[15];

strncpy(str2,str1,6);

printf("%s",str2);

}

1. W.A.P. to implement string using strlwr().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[10]="PRASAD";

strlwr(str);

printf("%s",str);

}

1. W.A.P. to implement string using strupr().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="mobiveil technology";

strupr(str);

printf("%s",str);

}

1. W.A.P. to implement string using strlen().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[10]="Prasad";

int p,q;

p=strlen(str);

q=sizeof(str);

printf("%d\n%d\n",p,q);

return 0;

}

1. W.A.P. to implement string using strchr().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[50]="Mobiveil Technology Pvt Ltd";

char \*str2;

char ch=' ';

str2=strchr(str,ch);

printf("%s",str2);

}

1. W.A.P. to implement string using strdup().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Mobiveil Technology,Hyderabad";

char \*sdup;

sdup=strdup(str1);

printf("%s",sdup);

}

1. W.A.P. to implementstring using strcmp()

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[10]="Prasad";

char str2[10]="Arjuna";

char str3[10]="Prasad";

int a,b;

if(sizeof(str1)==sizeof(str2))

{

a=strcmp(str1,str2);

b=strcmp(str1,str3);

}

printf("%d\n%d",a,b);

}

1. W.A.P. ti implement string using strcmpi().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[10]="Prasad";

char str2[10]="PRASAD";

char str3[10]="Prasad Kete";

char str4[10]="Pkete";

int a,b,c;

a=strcmpi(str1,str2);// returns 0

b=strcmpi(str1,str3);//returns negative value as length of str1 is less than str3

c=strcmpi(str1,str4);//returns positive value as length of str1 is less than str4

printf("%d\n%d\n%d",a,b,c);

}

1. W.A.P. to implement string using strrev().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[50]="Mobiveil Technology Pvt Ltd";

strrev(str);

printf("%s",str);

}

1. W.A.P. to implement string using strnset().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Mobiveil Technology,Hyderabad";

strnset(str1,'#',1);

printf("%s",str1);

}

1. W.A.P. to implement string without using strcpy().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[10]="Prasad",str2[10];

int i;

for(i=0;str1[i]!='\0';i++)

{

str2[i]=str1[i];

}

printf("%s",str2);

}

1. W.A.P. to implent string without using strcat().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str1[]="Prasad",str2[]=" is Embedded Developer",str3[50];

int i,j=0;

for(i=0;str1[i]!='\0';i++)

{

str3[j]=str1[i];

j++;

}

i=0;

for(i=0;str2[i]!='\0';i++)

{

str3[j]=str2[i];

j++;

}

printf("%s",str3);

}

1. W.A.P. to implement string without using strlen()

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="mobiveil Technology India,Hyderabad";

int i;

//First Logic

for(i=0;str[i]!='\0';i++);

printf("Length=%d",i);

//Second Logic

while(str[i]!='\0')

{

i++;

}

printf("Length=%d",i);

}

1. W.A.P. to implement string without using strupr()

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="prasad";

int i;

for(i=0;str[i]!='\0';i++)

{

if(str[i]>='a' && str[i]<='z')

{

str[i]=str[i]-32;

}

}

printf("Converted upper string=%s",str);

}

1. W.A.P. to implement string without using strlwr()

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="PRASAD";

int i;

for(i=0;str[i]!='\0';i++)

{

if(str[i]>='A' && str[i]<='Z')

{

str[i]=str[i]+32;

}

}

printf("Converted upper string=%s",str);

}

1. W.A.P. to implement string without strrev().

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char str[]="Tom & Jerry";

char str2[50];

int i,j,count=0;

while(str[count]!='\0')

count++;

j=count-1;

for(i=0;i<count;i++)

{

str2[i]=str[j];

j--;

}

str2[i]='\0';

printf("%s",str2);

}

1. W.A.P. to implement memchr() .

Ans:-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char \*str="Mobiveil is working on Semiconductor chip";

char \*ret=NULL;

ret=memchr(str,' ',strlen(str));

printf("%s",ret);

}

1. W.A.P. to implement memchr() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

void \*user\_memchr(const void \*str,int c,size\_t byte);

void \*user\_memchr(const void \*str,int c,size\_t byte)

{

unsigned char \*ptr=(unsigned char\*)str;

unsigned char ch=(unsigned char)c;

unsigned char \*p\_temp=NULL;

while(str!=NULL && byte--)

{

if(\*ptr!=ch)

{

ptr++;

}

else

{

p\_temp=ptr;

break;

}

}

return p\_temp;

}

int main()

{

char str[100],ch,\*ret;

printf("Enter string=");

gets(str);

printf("Enter character to be search=");

scanf("%c",&ch);

ret=user\_memchr(str,ch,strlen(str));

printf("String after memchr=%s\n",ret);

return 0;

}

1. W.A.P. to implement memcmp() & memcmpi().

Ans:-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char \*str1="mobiveil is semiconductor";

char \*str2="MOBIVEIL IS SEMICONDUCTOR";

int ret;

int ret2;

ret=memcmp(str1,str2,33);

printf("%d\n",ret);

if(ret==0)

{

printf("Both string are same\n");

}

else if(ret>0)

{

printf("Second string is less than first\n");

}

else

{

printf("First string is less than second\n");

}

ret2=memicmp(str1,str2,33);

if(ret2==0)

{

printf("Both string are same\n");

}

else if(ret2>0)

{

printf("Second string is less than first\n");

}

else

{

printf("First string is less than second\n");

}

}

1. W.A.P. to implement memcmp() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

int user\_memcmp(const void \*str1,const void \*str2,size\_t byte);

int user\_memcmp(const void \*str1,const void \*str2,size\_t byte)

{

char \*ptr1=(char\*)str1;

char \*ptr2=(char\*)str2;

int i;

for(i=0;i<byte;i++)

{

if(ptr1[i]==ptr2[i])

{

return 0;

}

if(ptr1[i]>ptr2[i])

{

return 1;

}

if(ptr1[i]<ptr2[i])

{

return -1;

}

}

}

int main()

{

char str1[100],str2[100];

int res;

printf("Enter String 1st=");

gets(str1);

printf("Enter String 2nd=");

gets(str2);

res=user\_memcmp(str1,str2,strlen(str2));

printf("Result=%d\n",res);

if(res==0)

{

printf("Both Strings are same\n");

}

if(res==1)

{

printf("String 1st is greater than String 2nd\n");

}

if(res==-1)

{

printf("String 1st is lesser than String 2nd\n");

}

return 0;

}

1. W.A.P. to implement memcmpi() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

int user\_memcmpi(const void \*str1,const void \*str2,size\_t byte);

int user\_memcmpi(const void \*str1,const void \*str2,size\_t byte)

{

char \*ptr1=(char\*)str1;

char \*ptr2=(char\*)str2;

int i;

if(byte==strlen(str1))

{

return 0;

}

if(byte<strlen(str1))

{

return 1;

}

if(byte>strlen(str1))

{

return -1;

}

}

int main()

{

char str1[100],str2[100];

int res;

printf("Enter String 1st=");

gets(str1);

printf("Enter String 2nd=");

gets(str2);

res=user\_memcmpi(str1,str2,strlen(str2));

printf("Result=%d\n",res);

if(res==0)

{

printf("Both Strings are same\n");

}

if(res==1)

{

printf("String 1st is greater than String 2nd\n");

}

if(res==-1)

{

printf("String 1st is lesser than String 2nd\n");

}

return 0;

}

1. W.A.P. to implement memcpy().

Ans:-

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int main()

{

char \*str1="Prasad";

char \*str2="Kete";

memcpy(str1,str2,strlen(str2));

printf("%s",str1);

}

1. W.A.P. to implement memcpy() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

void user\_memcpy(void \*dest,void \*src,size\_t byte);

void user\_memcpy(void \*dest,void \*src,size\_t byte)

{

char \*s\_src=(char\*)src;

char \*d\_des=(char\*)dest;

int i;

for(i=0;i<byte;i++)

{

d\_des[i]=s\_src[i];

}

}

int main()

{

char str1[100],str2[100];

printf("Enter string 1st=");

gets(str1);

user\_memcpy(str2,str1,strlen(str1));

printf("String 2 after memcpy=");

puts(str2);

return 0;

}

1. W.A.P. to implement memmove().

Ans:-

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int main()

{

char \*str1="Prasad";

char \*str2="Kete";

memmove(str1,str2,strlen(str2));

printf("%s",str1);

}

1. W.A.P. to implement memmove() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

void user\_memmove(void \*dest,void \*src,size\_t byte);

void user\_memmove(void \*dest,void \*src,size\_t byte)

{

char \*s\_src=(char\*)src;

char \*d\_des=(char\*)dest;

char \*temp[byte];

int i;

for(i=0;i<byte;i++)

{

temp[i]=s\_src[i];

}

for(i=0;i<byte;i++)

{

d\_des[i]=temp[i];

}

}

int main()

{

char str1[100],str2[100];

printf("Enter string 1st=");

gets(str1);

user\_memmove(str2,str1,strlen(str1));

printf("String 2 after memmove=");

puts(str2);

return 0;

}

1. W.A.P. to implement memset().

Ans:-

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char \*str="Prasad is Embedded Enginner";

printf("Before memset=%s\n",str);

memset(str,'i',10\*sizeof(char));

printf("After memset=%s\n",str);

}

1. W.A.P. to implement memset() by own logic.

Ans:-

#include<stdio.h>

#include<string.h>

void \*user\_memset(const void \*str,int ch,size\_t byte);

void \*user\_memset(const void \*str,int che,size\_t byte)

{

char \*ptr=(char\*)str;

unsigned char ch=(unsigned char)che;

while(byte--)

{

\*ptr=ch;

\*ptr++;

}

return str;

}

int main()

{

char str[100],ch,\*ret;

printf("Enter string=");

gets(str);

printf("Enter character to be set=");

scanf("%c",&ch);

ret=user\_memset(str,ch,6);

printf("String after memset=%s\n",ret);

return 0;

}

1. W.A.P. to convert decimal to binary.

Ans:-

#include<stdio.h>

int main()

{

long num,rem,binary=0,temp=1;

printf("Enter number=");

scanf("%d",&num);

while(num!=0)

{

rem=num%2;

num=num/2;

binary=binary+rem\*temp;

temp=temp\*10;

}

printf("%ld",binary);

}

1. W.A.P. to convert decimal to octal.

Ans:-

#include<stdio.h>

int main()

{

long num,rem,octal=0,temp=1;

printf("Enter number=");

scanf("%d",&num);

while(num!=0)

{

rem=num%8;

num=num/8;

octal=octal+rem\*temp;

temp=temp\*10;

}

printf("%ld",octal);

}

1. W.A.P. to convert decimal to hex.

Ans:-

#include<stdio.h>

int main()

{

long num,rem,temp=1,i=0,j;

char hex\_arra[32];

printf("Enter number=");

scanf("%d",&num);

while(num!=0)

{

rem=num%16;

if(rem<10)

{

hex\_arra[i]=rem+48;

i++;

}

else

{

hex\_arra[i]=rem+55;

i++;

}

num=num/16;

}

for(j=i-1;j>=0;j--)

{

printf("%c",hex\_arra[j]);

}

}

1. W.A.P. to convert binary to decimal.

Ans:-

#include<stdio.h>

int main()

{

int binary,rem,temp=1,dec=0;

printf("Enter binary number=");

scanf("%d",&binary);

while(binary!=0)

{

rem=binary%10;

binary=binary/10;

dec=dec+rem\*temp;

temp=temp\*2;

}

printf("%d",dec);

return 0;

}

1. W.A.P. to convert binary to octal.

Ans:-

#include<stdio.h>

int main()

{

int binary,dec=0,octal=0,temp=1,rem;

printf("Enter binary number=");

scanf("%d",&binary);

while(binary!=0)

{

rem=binary%10;

binary=binary/10;

octal=octal+rem\*temp;

temp=temp\*2;

}

while(dec!=0)

{

rem=dec%8;

dec=dec/8;

octal=octal+rem\*temp;

temp=temp\*8;

}

printf("%o",octal);

}

1. W.A.P. to convert binary to hex.

Ans:-

#include<stdio.h>

int main()

{

int binary,dec=0,temp=1,rem,i,j;

char hex\_array[32];

printf("Enter binary number=");

scanf("%d",&binary);

while(binary!=0)

{

rem=binary%10;

binary=binary/10;

dec=dec+rem\*temp;

temp=temp\*2;

}

while(dec!=0)

{

rem=dec%16;

if(rem<10)

{

hex\_array[i]=rem+48;

i++;

}

else

{

hex\_array[i]=rem+55;

i++;

}

dec=dec/16;

}

for(j=i-1;j>=0;j--)

{

printf("%c",hex\_array[j]);

}

}

1. W.A.P. to convert octal to decimal.

Ans:-

#include<stdio.h>

int main()

{

int dec=0,octal,temp=1,rem;

printf("Enter octal=");

scanf("%d",&octal);

while(octal!=0)

{

rem=octal%10;

octal=octal/10;

dec=dec+rem\*temp;

temp=temp\*8;

}

printf("%d",dec);

}

1. W.A.P. to convert octal to binary.

Ans:-

#include<stdio.h>

int main()

{

int octal,dec=0,rem,temp=1,binary=0;

printf("Enter octal=");

scanf("%d",&octal);

while(octal!=0)

{

rem=octal%10;

octal=octal/10;

dec=dec+rem\*temp;

temp=temp\*8;

}

printf("%d\n",dec);

rem=0;

temp=1;

while(dec!=0)

{

rem=dec%2;

dec=dec/2;

binary=binary+rem\*temp;

temp=temp\*10;

}

printf("%d",binary);

}

1. W.A.P. to convert Octal to hex.

Ans:-

#include<stdio.h>

int main()

{

int octal,dec=0,rem,temp=1,i,j;

char hex\_array[32];

printf("Enter octal=");

scanf("%d",&octal);

while(octal!=0)

{

rem=octal%10;

octal=octal/10;

dec=dec+rem\*temp;

temp=temp\*8;

}

printf("%d\n",dec);

rem=0;

temp=1;

while(dec!=0)

{

rem=dec%16;

if(rem>10)

{

hex\_array[i]=rem+48;

i++;

}

else

{

hex\_array[i]=rem+55;

i++;

}

dec=dec/16;

}

for(j=i-1;j>=0;j--)

{

printf("%c",hex\_array[j]);

}

}

1. W.A.P. to convert hex to decimal.

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char hex[32];

int dec=0,rem=0,temp=1,i,length;

printf("Enter hex value=");

scanf("%s",hex);

length=strlen(hex);

printf("%d\n",length);

for(i=length-1;i>=0;i--)

{

if(hex[i]>='0' && hex[i]<='9')

{

rem=hex[i] - '0';

}

else

{

rem=hex[i]-'A' + 10;

}

dec=dec+rem\*temp;

temp=temp\*16;

}

printf("%d",dec);

}

1. W.A.P. to convert hex to binary

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char hex[32];

int dec=0,rem=0,temp=1,i,binary=0,length;

printf("Enter hex value=");

scanf("%s",hex);

length=strlen(hex);

printf("%d\n",length);

for(i=length-1;i>=0;i--)

{

if(hex[i]>='0' && hex[i]<='9')

{

rem=hex[i] - '0';

}

else

{

rem=hex[i] - 'A' +10;

}

dec=dec+rem\*temp;

temp=temp\*16;

}

printf("%d\n",dec);

rem=0;

temp=1;

while(dec!=0)

{

rem=dec%2;

dec=dec/2;

binary=binary+rem\*temp;

temp=temp\*10;

}

printf("%d",binary);

}

1. W.A.P. to convert hex to octal.

Ans:-

#include<stdio.h>

#include<string.h>

int main()

{

char hex[32];

int dec=0,rem=0,temp=1,i,octal=0,length;

printf("Enter hex value=");

scanf("%s",hex);

length=strlen(hex);

printf("%d\n",length);

for(i=length-1;i>=0;i--)

{

if(hex[i]>='0' && hex[i]<='9')

{

rem=hex[i] -'0';

}

else

{

rem=hex[i] -'A' +10;

}

dec=dec+rem\*temp;

temp=temp\*16;

}

printf("%d\n",dec);

rem=0;

temp=1;

while(dec!=0)

{

rem=dec%8;

dec=dec/8;

octal=octal+rem\*temp;

temp=temp\*8;

}

printf("%o",octal);

}

1. Implement malloc() function.

Ans:\_

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*p=NULL;

int i;

p=(int\*)malloc(20\*sizeof(int));

if(p==NULL)

{

printf("Memory is not allocated\n");

}

else

{

printf("Memory is allocated using malloc() successfully\n");

for(i=0;i<10;i++)

{

p[i]=i+1;

}

for(i=0;i<10;i++)

{

printf("%d\t",p[i]);

}

}

free(p);

}

1. Implement calloc() function.

Ans:-

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*ptr=NULL;

int i;

ptr=(int\*)calloc(10,sizeof(int));

if(ptr==NULL)

{

printf("Memory is not allocated");

}

else

{

printf("Memory is allocated succesfully using calloc() function\n");

for(i=0;i<10;i++)

{

ptr[i]=i+1;

}

for(i=0;i<10;i++)

{

printf("%d\n",ptr[i]);

}

}

free(ptr);

}

1. Implement realloc() function.

Ans:-

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*ptr=NULL;

int n=5;

int i;

ptr=(int\*)malloc(n\*sizeof(int));

if(ptr==NULL)

{

printf("Memory is not allocated\n");

}

else

{

printf("Memory is allocated successfully using malloc() \n");

for(i=0;i<n;i++)

{

ptr[i]=i+1;

}

for(i=0;i<n;i++)

{

printf("%d\t",ptr[i]);

}

}

printf("\n");

n=10;

ptr=realloc(ptr,n\*sizeof(int));

printf("Memory is allocated successfully using realloc() \n");

for(i=6;i<=n;i++)

{

ptr[i]=i+1;

}

for(i=6;i<=n;i++)

{

printf("%d\t",ptr[i]);

}

free(ptr);

}

1. W.A.P. to implement atof() function.

Ans:-

/\*Convert string to float\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str="3.14";

float num;

num=atof(str);

printf("%f",num);

}

1. W.A.P. to implement atoi() function.

Ans:-

/\*Convert string to integer\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str="31457";

int num;

num=atoi(str);

printf("%d",num);

}

1. W.A.P. to implement atoll() function

Ans:-

/\*Convert string to long\*/

/\*Forconverting string to long ,maximum array size is 10\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str="1234567891";

long num;

num=atol(str);

printf("%ld",num);

}

1. W.A.P. to implement gcvt() function.

Ans:-

/\*Convert Float value to String\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

double num=3.5448541245;

char \*str1;

gcvt(num,8,str1);

printf("%s\n",str1);

}

1. W.A.P. to implement itoa() function.

Ans:-

/\*Convert integer to string\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

int num=10;

char \*str1;

itoa(num,str1,10);

printf("Binary value=%s\n",str1);

itoa(num,str1,2);

printf("Binary value=%s\n",str1);

itoa(num,str1,16);

printf("Hexadecimal value=%s\n",str1);

itoa(num,str1,8);

printf("Octal value=%s\n",str1);

}

1. W.A.P. to implement ltoa()

Ans:-

/\*Convert long to string\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

long num=1000000000;

char \*str1;

ltoa(num,str1,10);

printf("Binary value=%s\n",str1);

ltoa(num,str1,2);

printf("Binary value=%s\n",str1);

ltoa(num,str1,16);

printf("Hexadecimal value=%s\n",str1);

ltoa(num,str1,8);

printf("Octal value=%s\n",str1);

}

1. W.A.P. to implement strtod() function.

Ans:-

/\*Convert string to double\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str1="3.14";

char \*end;

double num;

num=strtod(str1,&end);

printf("%f",num);

}

1. W.A.P. to implement strtol() function.

Ans:-

/\*Convert String to long\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str1="2021000000 is the best year";

char \*ptr;

long num;

num=strtol(str1,&ptr,10);

printf("Decimal value=%ld\n",num);

if(\*ptr)

{

printf("%s\n",ptr);

}

else

{

printf("Null pointer");

}

}

1. W.A.P. to implement strtoul() function.

Ans:-

/\*Convert string to unsigned long\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

char \*str1="2021000000 is the best year";

unsigned long num;

char \*ptr;

num=strtoul(str1,&ptr,10);

printf("%lu\n",num);

if(\*ptr)

{

printf("%s",ptr);

}

else

{

printf("NUll POINTER");

}

}

1. W.A.P. to implement ultoa() function.

Ans:-

/\*convert long to string\*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

unsigned long num=10;

char \*str;

ultoa(num,str,2);

printf("%s",str);

}

1. W.A.P. using #define & #include

Ans:-

#include<stdio.h>

#define A 10

#define B 15

int main()

{

int c;

c=A+B;

printf("%d",c);

return 0;

}

1. W.A.P. using #undef or redefine a macro.

Ans:-

#include<stdio.h>

#define A 50

#define B 25

#undef B //First way to undef

int main()

{

int c;

#undef A //Second way to undef

#define A 25 //redefine macro

c=A\*A;

printf("%d",c);

return ;

}

1. W.A.P. using conditional macros(#if,#else,#elif,#endif).

Ans:-

#include<stdio.h>

#define POSITIVE 1

#define NEGATIVE -1

int main()

{

#if POSITIVE>0

printf("Number started positively\n");

#elif NEGATIVE<0

printf("Number started negatively\n");

#else

printf("There is character\n");

#endif

return 0;

}

1. W.A.P. using #ifdef & #ifndef check whether macro is defined or not.

Ans:-

#include<stdio.h>

#define NUM 10

int main()

{

#ifdef NUM

printf("NUM Macro is present\n");

#else

printf("NUM Macro is not present\n");

#endif

#ifndef MAX

printf("MAX Macro is not present\n");

#else

printf("MAX Macro is present\n");

#endif

}

1. W.A.P. for #line macro

Ans:-

#include<stdio.h>

int main()

{

printf("Addition of two numbers=\n");

printf("%d\n",15+25);

#line 6 // Resetting the line

printf("%d\n",15+15);

}

1. W.A.P. for # & ## pre-processor operator.

Ans:-

#include<stdio.h>

#define STR(x) #x

#define STR\_CAT(x,y) x##y

int main()

{

printf(STR(Mobiveil is semiconductor Compamny));

printf("\n");

printf("%d\n",STR\_CAT(20,21));

return 0;

}

1. W.A.P. to print error in program using #error.

Ans:-

#include<stdio.h>

#define A 30

#define B 25

#define C A+B

int main()

{

#ifndef C

#error Please declare variable to store addition

#else

printf("%d",C);

#endif

return 0;

}

1. W.A.P. to find line number i.e.\_LINE\_ macros

Ans:-

#include<stdio.h>

int main()

{

printf("hello\n");

printf("Good Afternoon\n");

printf("Line=%d\n",\_\_LINE\_\_); //Output = 30 if line is 30

return 0;

}

1. W.A.P. to find function name i.e.\_FUNCTION\_ macros

Ans:-

/\* \_\_func\_\_ & \_\_FUNCTION\_\_ are same\*/

#include<stdio.h>

void fun1(void);

void fun2(void);

void fun1()

{

printf("Calling function is=%s\n",\_\_func\_\_);

}

void fun2()

{

printf("Calling function is=%s\n",\_\_func\_\_);

}

int main()

{

printf("Called function is=%s\n",\_\_func\_\_);

fun1();

fun2();

}

1. W.A.P. to find file name i.e. \_FILE\_ macros

Ans:-

#include<stdio.h>

int main()

{

printf("Current file name=%s",\_\_FILE\_\_);

return 0;

}

1. W.A.P. to set,clear & toggle a bit using macros

Ans:-

#include<stdio.h>

#define SET(x,y) (x |= 1<<y)

#define CLEAR(x,y) (x &= ~(1<<y))

#define TOGGLE(x,y) (x ^= (1<<y))

int main()

{

int p=10;

int q=2;

SET(p,q);

printf("Set bit of 10 by 2 is=%d\n",p);

int m=10;

int n=3;

CLEAR(m,n);

printf("Result after clear 2 bit of 10=%d\n",m);

int a=10;

int b=2;

TOGGLE(a,b);

printf("Result after toggle 2 bit of 10=%d\n",a);

}

1. W.A.P. to set nth bit to zero.

Ans:-

#include<stdio.h>

#define SET(x,y) (x &= ~(1<<y))

int main()

{

int p=10;

int q=1;

SET(p,q);

printf("Result after 3 bit to zero of 10=%d\n",p);

return 0;

}

1. W.A.P. to execute a particular function using maros.

Ans:-

#include<stdio.h>

#define ADD(A,B) A+B

int main()

{

int r;

int p=10,q=20;

r=ADD(p,q);

printf("%d",r);

return 0;

}

1. W.A.P. to define array size using macro

Ans:-

#include<stdio.h>

#define MAX 5

int main()

{

int arr[MAX]={1,2,3,4,5};

int i;

for(i=0;i<MAX;i++)

{

printf("%d\t",arr[i]);

}

}

1. W.A.P. to execute program without main function

Ans:-

#include<stdio.h>

#define START main

int START()

{

printf("Program without main function");

}

1. W.A.P. for file handling functions.

Ans:-

#include<stdio.h>

int main()

{

FILE \*file;

char ch;

file=fopen("E:\\File.txt","w"); //create & Open File

if(file=fopen("E:\\File.txt","w"))

{

printf("File is created succesfully\n");

}

fprintf(file,"Mobiveil Company Inormation"); //Write to file

file=fopen("E:\\File.txt","a"); //append to File

fprintf(file,"\nChennai is head office of Mobiveil in INDIA");

file=fopen("E:\\File.txt","r"); //read to File

while(1)

{

ch=fgetc(file);

if(ch==EOF)

break;

else

printf("%c",ch);

}

file=fopen("E:\\File.txt","a"); //create & Open File & use fputs

fputs("\nMobiveil is Servicce based company also",file);

fclose(file);

if(fclose(file))

printf("File successfully closed");

}

1. W.A.P. to Set a bit

Ans:-

#include<stdio.h>

void set\_bit(int x,int y);

void set\_bit(int x,int y)

{

x |= 1<<y;

printf("%d",x);

}

int main()

{

int a,b;

printf("Enter a number=");

scanf("%d",&a);

printf("Enter a bit which is set=");

scanf("%d",&b);

set\_bit(a,b);

return 0;

}

1. W.A.P. to Clear a bit

Ans:-

#include<stdio.h>

void clear\_bit(int x,int y);

void clear\_bit(int x,int y)

{

x &=~(1<<y);

printf("%d",x);

}

int main()

{

int a,b;

printf("Enter number=");

scanf("%d",&a);

printf("Enter bit for clear=");

scanf("%d",&b);

clear\_bit(a,b);

return 0;

}

1. Toggle a bit

Ans:-

#include<stdio.h>

void toggle\_bit(int x,int y);

void toggle\_bit(int x,int y)

{

x ^= 1<<y;

printf("%d",x);

}

int main()

{

int a,b;

printf("Enter number=");

scanf("%d",&a);

printf("Enter bit for toggle=");

scanf("%d",&b);

toggle\_bit(a,b);

return 0;

}

1. Shift bit to right

Ans:-

#include<stdio.h>

void shift\_right(int x,int y);

void shift\_right(int x,int y)

{

x=x>>y;

printf("%d",x);

}

int main()

{

int a,b;

printf("Enter number=");

scanf("%d",&a);

printf("Enter a bit for right shift=");

scanf("%d",&b);

shift\_right(a,b);

return 0;

}

1. Shift bit to left

Ans:-

#include<stdio.h>

void shift\_left(int x,int y);

void shift\_left(int x,int y)

{

x=x<<y;

printf("%d",x);

}

int main()

{

int a,b;

printf("Enter number=");

scanf("%d",&a);

printf("Enter bit for left shift=");

scanf("%d",&b);

shift\_left(a,b);

return 0;

}

1. W.A.P. for enumeration.

Ans:\_

#include<stdio.h>

enum week{Mon,Tue,Wed,Thu,Fri,Sat,Sun};

enum month{Jan=1,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec};

int main()

{

printf("%d %d %d %d %d %d %d\n",Mon,Tue,Wed,Thu,Fri,Sat,Sun);

printf("%d %d %d %d %d %d %d %d %d %d %d d\n",Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec);

return 0;

}

1. W.A.P. for typedef.

Ans:-

#include<stdio.h>

int main()

{

typedef int dec; //Give integer name as dec

typedef char alpha;//Give character name as alpha

typedef float exp;//Give float name as exp

dec a=10;

alpha b='A';

exp m=10.143;

printf("%d\n",a);

printf("%c\n",b);

printf("%f\n",m);

return 0;

}

1. W.A.P. for typecasting.

Ans:-

#include<stdio.h>

int main()

{

char m='A';

int x;

printf("Before typecasting m=%c\n",m);

x=(int)m;

printf("After typecasting m=%d\n",x);

return 0;

}

1. W.A.P. to create structure.

Ans:-

#include<stdio.h>

struct empolyee\_id

{

char name[20];

char post[20];

char company[20];

char em\_id[7];

}Prasad,Roopa,Supriya,Nagarjuna,Sai;

int main()

{

// /# Prasad Infromation #/

printf("Enter 1st Employee name=");

scanf("%s",Prasad.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Prasad.post);

printf("Enter 1st Employee company name=");

scanf("%s",Prasad.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Prasad.em\_id);

///#Roopa Information#/

printf("Enter 2nd Employee name=");

scanf("%s",Roopa.name);

printf("Enter 2nd Employee Designation=");

scanf("%s",Roopa.post);

printf("Enter 2nd Employee company name=");

scanf("%s",Roopa.company);

printf("Enter 2nd Emplpoyee id=");

scanf("%s",Roopa.em\_id);

///#Supriya Information#/

printf("Enter 3rd Employee name=");

scanf("%s",Supriya.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Supriya.post);

printf("Enter 1st Employee company name=");

scanf("%s",Supriya.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Supriya.em\_id);

///#Nagarjuna Information#/

printf("Enter 1st Employee name=");

scanf("%s",Nagarjuna.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Nagarjuna.post);

printf("Enter 1st Employee company name=");

scanf("%s",Nagarjuna.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Nagarjuna.em\_id);

// /#Sai Information#/

printf("Enter 1st Employee name=");

scanf("%s",Sai.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Sai.post);

printf("Enter 1st Employee company name=");

scanf("%s",Sai.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Sai.em\_id);

/\*Prasad.name="Prasad";

Prasad.post="Embedded Software Develepr";

Prasad.company="Mobiveil";

Prasad.em\_id="EN801";

Roopa.name="Roopa";

Roopa.post="Embedded Software Develepr";

Roopa.company="Mobiveil";

Roopa.em\_id="EN802";

Supriya.name="Supriya";

Supriya.post="Embedded Software Develepr";

Supriya.company="Mobiveil";

Supriya.em\_id="EN803";

Nagarjuna.name="Nagarjuna";

Nagarjuna.post="Embedded Software Develepr";

Nagarjuna.company="Mobiveil";

Nagarjuna.em\_id="EN804";

Sai.name="Sai";

Sai.post="Embedded Software Develepr";

Sai.company="Mobiveil";

Sai.em\_id="EN805";\*/

printf("1st Employee Information\n");

printf("Name:-%s\n",Prasad.name);

printf("Designation:-%s\n",Prasad.post);

printf("Company:-%s\n",Prasad.company);

printf("Employee ID:-%s\n",Prasad.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("2nd Employee Information\n");

printf("Name:-%s\n",Roopa.name);

printf("Designation:-%s\n",Roopa.post);

printf("Company:-%s\n",Roopa.company);

printf("Employee ID:-%s\n",Roopa.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("3rd Employee Information\n");

printf("Name:-%s\n",Supriya.name);

printf("Designation:-%s\n",Supriya.post);

printf("Company:-%s\n",Supriya.company);

printf("Employee:-%s\n",Supriya.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("4th Employee Information\n");

printf("Name:-%s\n",Nagarjuna.name);

printf("Designation:-%s\n",Nagarjuna.post);

printf("Company:-%s\n",Nagarjuna.company);

printf("Employee ID%s\n",Nagarjuna.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("5th Employee Information\n");

printf("Name:-%s\n",Sai.name);

printf("Designation:-%s\n",Sai.post);

printf("Company:-%s\n",Sai.company);

printf("Employee ID:-%s\n",Sai.em\_id);

}

1. W.A.P. for union implement..

Ans:-

#include<stdio.h>

union empolyee\_id

{

char name[20];

char post[20];

char company[20];

char em\_id[7];

}Prasad,Roopa,Supriya,Nagarjuna,Sai;

int main()

{

// /# Prasad Infromation #/

printf("Enter 1st Employee name=");

scanf("%s",Prasad.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Prasad.post);

printf("Enter 1st Employee company name=");

scanf("%s",Prasad.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Prasad.em\_id);

///#Roopa Information#/

printf("Enter 2nd Employee name=");

scanf("%s",Roopa.name);

printf("Enter 2nd Employee Designation=");

scanf("%s",Roopa.post);

printf("Enter 2nd Employee company name=");

scanf("%s",Roopa.company);

printf("Enter 2nd Emplpoyee id=");

scanf("%s",Roopa.em\_id);

///#Supriya Information#/

printf("Enter 3rd Employee name=");

scanf("%s",Supriya.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Supriya.post);

printf("Enter 1st Employee company name=");

scanf("%s",Supriya.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Supriya.em\_id);

///#Nagarjuna Information#/

printf("Enter 1st Employee name=");

scanf("%s",Nagarjuna.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Nagarjuna.post);

printf("Enter 1st Employee company name=");

scanf("%s",Nagarjuna.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Nagarjuna.em\_id);

// /#Sai Information#/

printf("Enter 1st Employee name=");

scanf("%s",Sai.name);

printf("Enter 1st Employee Designation=");

scanf("%s",Sai.post);

printf("Enter 1st Employee company name=");

scanf("%s",Sai.company);

printf("Enter 1st Emplpoyee id=");

scanf("%s",Sai.em\_id);

/\*Prasad.name="Prasad";

Prasad.post="Embedded Software Develepr";

Prasad.company="Mobiveil";

Prasad.em\_id="EN801";

Roopa.name="Roopa";

Roopa.post="Embedded Software Develepr";

Roopa.company="Mobiveil";

Roopa.em\_id="EN802";

Supriya.name="Supriya";

Supriya.post="Embedded Software Develepr";

Supriya.company="Mobiveil";

Supriya.em\_id="EN803";

Nagarjuna.name="Nagarjuna";

Nagarjuna.post="Embedded Software Develepr";

Nagarjuna.company="Mobiveil";

Nagarjuna.em\_id="EN804";

Sai.name="Sai";

Sai.post="Embedded Software Develepr";

Sai.company="Mobiveil";

Sai.em\_id="EN805";\*/

printf("1st Employee Information\n");

printf("Name:-%s\n",Prasad.name);

printf("Designation:-%s\n",Prasad.post);

printf("Company:-%s\n",Prasad.company);

printf("Employee ID:-%s\n",Prasad.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("2nd Employee Information\n");

printf("Name:-%s\n",Roopa.name);

printf("Designation:-%s\n",Roopa.post);

printf("Company:-%s\n",Roopa.company);

printf("Employee ID:-%s\n",Roopa.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("3rd Employee Information\n");

printf("Name:-%s\n",Supriya.name);

printf("Designation:-%s\n",Supriya.post);

printf("Company:-%s\n",Supriya.company);

printf("Employee:-%s\n",Supriya.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("4th Employee Information\n");

printf("Name:-%s\n",Nagarjuna.name);

printf("Designation:-%s\n",Nagarjuna.post);

printf("Company:-%s\n",Nagarjuna.company);

printf("Employee ID%s\n",Nagarjuna.em\_id);

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("5th Employee Information\n");

printf("Name:-%s\n",Sai.name);

printf("Designation:-%s\n",Sai.post);

printf("Company:-%s\n",Sai.company);

printf("Employee ID:-%s\n",Sai.em\_id);

}

1. Write a program to print “Embedded World”.

Ans:-

#include<stdio.h>

int main()

{

printf("Embedded World");

return 0;

}

1. Write a program to take int, char, float, long, double ,short, unsigned variable and print it with its size.

Ans:-

#include<stdio.h>

int main()

{

int a;

char b;

float c;

long d;

double e;

short f;

unsigned g;

printf("size of int=%d\n",sizeof(a));

printf("size of char=%d\n",sizeof(b));

printf("size of float=%d\n",sizeof(c));

printf("size of long=%ld\n",sizeof(d));

printf("size of double=%d\n",sizeof(e));

printf("size of short=%d\n",sizeof(f));

printf("size of unsigned=%d\n",sizeof(g));

return 0;

}

1. Find the ASCII value of character.

Ans:-

#include<stdio.h>

int main()

{

char ch;

printf("Enter Character=");

scanf("%c",&ch);

printf("ASCII value of %c is=%d",ch,ch);

return 0;

}

1. Printf ASCII table character.

Ans:-

#include<stdio.h>

int main()

{

int i;

char ch;

for(i=0;i<256;i++)

{

printf("%c\t",ch);

ch++;

}

return 0;

}

1. Swapping of two numbers.

Ans:-

#include<stdio.h>

int main()

{

int a;

int b;

int temp;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

printf("Number before swapping\n");

printf("a=%d\nb=%d\n",a,b);

temp=a;

a=b;

b=temp;

printf("Number after swapping\n");

printf("a=%d\nb=%d\n",a,b);

return 0;

}

1. Swapping of two two number without using third variable.

Ans:-

#include<stdio.h>

int main()

{

int a;

int b;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

printf("Number before swapping\n");

printf("a=%d\nb=%d\n",a,b);

/\*First Method/

a=a+b;

b=a-b;

a=a-b;

printf("Number after swapping\n");

printf("a=%d\nb=%d\n",a,b);

/\*Second Method\*/

a=a\*b;

b=a/b;

a=a/b;

printf("Number after swapping\n");

printf("a=%d\nb=%d\n",a,b);

/\*Third Method\*/

a=a^b;

b=a^b;

a=a^b;

printf("Number after swapping\n");

printf("a=%d\nb=%d\n",a,b);

return 0;

}

1. Swap three numbers.

Ans:-

#include<stdio.h>

int main()

{

int x;

int y;

int z;

int temp;

printf("Enter first number=");

scanf("%d",&x);

printf("Enter second number=");

scanf("%d",&y);

printf("Enter third number=");

scanf("%d",&z);

printf("Number before swapping\n");

printf("x=%d\ny=%d\nz=%d\n",x,y,z);

temp=x;

x=y;

y=z;

z=temp;

printf("Number after swapping\n");

printf("x=%d\ny=%d\nz=%d\n",x,y,z);

return 0;

}

1. Swap two nibbles .

Ans:-

#include<stdio.h>

int main()

{

int num,temp;

printf("Enter num=");

scanf("%d",&num);

temp=((num & 0X0F)<<4 | (num & 0XF0)>>4);

printf("%d",temp);

return 0;

}

1. Perform ESCAPE sequence.

Ans:-

#include<stdio.h>

#include<stdlib.h>

#define a 10

int main()

{

//backspace

printf("Hello all\n");

printf("Hello\b all");

//new line

printf("Hello\n");

printf("Good evening");

//carriage return

printf("very Good Morning");

printf("\rHello ");

//horizontal tab

printf("1234\n");

printf("1\t2\t3\t4\t");

//vertical tab

printf("1234 ");

printf("\v1\v2\v3\v4");

//Null

printf("Hello,Good Evening\n");

printf("Hello \0 ,Good Evening");

//Double quote

printf("We all are \"Embedded Enginner\"\n");

//Single quote

printf("we worked at \'Mobiveil\'\n");

//Backslash

printf("c\\program files\\Orcad\\Setup");

//octal Constant

char \*a="\101";

printf("%s",a);

//Hexadecimal Constant

char \*a="\x41";

printf("%s",a);\*/

}

1. Write a program for Arithmetic Operator.

Ans:-

#include<stdio.h>

int main()

{

int a,b,result1,result2,result3,result4,result5,result6,result7,result8,result9;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

result1=a+b;

printf("Addition=%d\n",result1);

result2=a-b;

printf("Subtraction=%d\n",result2);

result3=a\*b;

printf("Multiplication=%d\n",result3);

result4=a/b;

printf("Division=%d\n",result4);

result5=a%b;

printf("Modulus=%d\n",result5);

result6=a++;

printf("Post Increment=%d\n",result6);

result7=++a;

printf("Pre-increment=%d\n",result7);

result8=++b;

printf("Pre-increment=%d\n",result8);

result9=b++;

printf("Post Increment=%d\n",result9);

return 0;

}

1. Write a program for Relational Operator.

Ans:-

#include<stdio.h>

int main()

{

int a;

int b;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

if(a>b)

{

printf("a is grater\n");

}

if(a<b)

{

printf("b is greater\n");

}

if(a>=b)

{

printf("a is greater than or equal to b\n");

}

if(a<=b)

{

printf("a is less than or equal to b\n");

}

if(a==b)

{

printf("a & b are equal\n");

}

if(a!=b)

{

printf("a is not equal to b\n");

}

else

{

printf("there is no relation between a & b\n");

}

return 0;

}

1. Write a program for Logical Operator.

Ans:-

#include<stdio.h>

int main()

{

int a;

int b;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

if(a>0 && b>0)

{

printf("Positive Number\n");

}

if(a<0 || b<0)

{

printf("Negative Number\n");

}

if(a!=b)

{

printf("Unequal Number\n");

}

return 0;

}

1. Write a program for Assignment operator.

Ans:-

#include<stdio.h>

int main()

{

int a,result1,result2,result3,result4,result5,result6,result7,result8,result9,result10,result11;

a=5;

result1=a;

printf("Equal to=%d\n",result1);

a+=5;

result2=a;

printf("Addition=%d\n",result2);

a-=5;

result3=a;

printf("Subtarction=%d\n",result3);

a/=5;

result4=a;

printf("Divide=%d\n",result4);

a%=5;

result5=a;

printf("Modulus=%d\n",result5);

a\*=5;

result6=a;

printf("Multipication=%d\n",result6);

a<<=1;

result7=a;

printf("Left Shift=%d\n",result7);

a>>=1;

result8=a;

printf("Right Shift=%d\n",result8);

a&=1;

result9=a;

printf("Bitwise AND=%d\n",result9);

a|=1;

result10=a;

printf("Bitwise OR=%d\n",result9);

a^=1;

result11=a;

printf("Bitwise XOR=%d\n",result11);

return 0;

}

1. Write a program for conditional operator.

Ans:-

#include<stdio.h>

int main()

{

int x;

int y;

int res;

printf("Enter x=");

scanf("%d",&x);

printf("Enter y=");

scanf("%d",&y);

res=(x>y) ? x : y ;

printf("%d",res);

return 0;

}

1. Write a program for special operator.

Ans:-

/\* "&" and "\*" are sepcial operator\*/

#include<stdio.h>

int main()

{

int a=10;

int \*p;

p=&a;

printf("%d\n%d\n",p,\*p);

return 0;

}

1. Write a program for Bitwise Operator.

Ans:-

#include<stdio.h>

int main()

{

int a,result1,result2,result3,result4;

printf("Enter a=");

scanf("%d",&a);

result1= a & 1;

printf("Bitwsise AND=%d\n",result1);

result2= a | 1;

printf("Bitwsise OR=%d\n",result2);

result3= a ^ 1;

printf("Bitwsise XOR=%d\n",result3);

result4= ~a;

printf("Complement=%d\n",result4);

return 0;

}

1. Write a program to check whether number is even or odd.

Ans:-

#include<stdio.h>

int main()

{

int num;

printf("Enter number=");

scanf("%d",&num);

if(num%2==0)

{

printf("%d is even number",num);

}

else

{

printf("%d is odd number",num);

}

return 0;

}

1. Write a program to check whether character is vowels or consonant.

Ans:-

#include<stdio.h>

int main()

{

char ch;

printf("Enter character=");

scanf("%c",&ch);

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

{

printf("%c is Vowel",ch);

}

else

{

printf("%c is Consonant",ch);

}

return 0;

}

1. C program to find largest & smallest number among three numbers.

Ans:-

#include<stdio.h>

int main()

{

int num1,num2,num3;

printf("Enter num1=");

scanf("%d",&num1);

printf("Enter num2=");

scanf("%d",&num2);

printf("Enter num3=");

scanf("%d",&num3);

if(num1==num2 && num1==num3 && num2==num3)

{

printf("All numbers are equal");

}

else if(num1>num2 && num1>num3)

{

printf("num1 is greater");

}

else if(num2>num3)

{

printf("num2 is greater");

}

else

{

printf("num3 is greater");

}

return 0;

}

1. C program to check whether year is Leap or not.

Ans:-

#include<stdio.h>

int main()

{

int year;

printf("Enter year=");

scanf("%d",&year);

if(year%4==0)

{

printf("%d Year is leap year",year);

}

else

{

printf("%d Year is not leap year",year);

}

return 0;

}

1. C program to check whether character is alphabet or not.

Ans:-

#include<stdio.h>

int main()

{

char ch;

printf("Enter character=");

scanf("%c",&ch);

if(ch>='a' && ch<='z' || ch>='A' && ch<='Z')

{

printf("%c is an alphabet",ch);

}

else

{

printf("%c is a number or symbol",ch);

}

return 0;

}

1. C program to check whether number is palindrome or not.

Ans:-

#include<stdio.h>

int main()

{

int num;

int sum=0,rem=0,temp;

printf("Enter num=");

scanf("%d",&num);

temp=num;

while(num!=0)

{

rem=num%10;

sum=sum\*10+rem;

num=num/10;

}

printf("%d\n",sum);

if(sum==temp)

{

printf("%d is palindrome",temp);

}

else

{

printf("%d is just a number",temp);

}

return 0;

}

1. C program to check whether number is prime or not.

Ans:-

#include<stdio.h>

int main()

{

int num,i,count=0;

printf("Enter number=");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

if(num%i==0)

{

count++;

}

}

if(count==2)

{

printf("%d is prime number",num);

}

else

{

printf("%d is not prime number",num);

}

return 0;

}

1. C program to check whether number is Armstrong or not.

Ans:-

#include<stdio.h>

int main()

{

int num;

int sum=0,rem=0,temp;

printf("Enter number=");

scanf("%d",&num);

temp=num;

while(num!=0)

{

rem=num%10;

sum=sum+rem\*rem\*rem;

num=num/10;

}

printf("%d\n",sum);

if(sum==temp)

{

printf("%d is Armstrong number",temp);

}

else

{

printf("%d is not Armstrong number",temp);

}

return 0;

}

1. Write a program to calculate sum of all natural numbers.

Ans:-

#include<stdio.h>

int main()

{

int num,i,sum=0;

printf("Enter num=");

scanf("%d",&num);

for(i=0;i<=num;i++)

{

sum+=i;

}

printf("%d",sum);

return 0;

}

1. C program to find factorial of number.

Ans:-

#include<stdio.h>

int main()

{

int num,i,fact=1;

printf("Enter number=");

scanf("%d",&num);

if(num<0)

{

printf("%d is negatve number & negative number doesn't have factorai",num);

}

else

{

printf("Factorial of %d are\n",num);

for(i=1;i<=num;i++)

{

fact\*=i;

}

printf("%d\t",fact);

}

return 0;

}

1. C program to generate multiplication table.

Ans:-

#include<stdio.h>

int main()

{

int num,i,table;

printf("Enter number=");

scanf("%d",&num);

printf("Multplication Table of %d is\n",num);

for(i=1;i<=10;i++)

{

table=i\*num;

printf("%d\n",table);

}

return 0;

}

1. C program to display Fibonacci sequence.

Ans:-

#include<stdio.h>

int main()

{

int num1=0,num2=1,num3=1,number,i;

printf("Enter numbr=");

scanf("%d",&number);

printf("Fibonacci Series upto %d is\n",number);

printf("%d\t",num1);

printf("%d\t",num2);

printf("%d\t",num3);

for(i=0;i<=number;i++)

{

num1=num2;

num2=num3;

num3=num1+num2;

printf("%d\t",num3);

}

return 0;

}

1. Display characters from A to Z & a to z using loop.

Ans:-

#include<stdio.h>

int main()

{

int cha;

for(cha=65;cha<=90;cha++)

{

printf("%c\t",cha);

}

printf("\n");

for(cha=97;cha<=120;cha++)

{

printf("%c\t",cha);

}

return 0;

}

1. C program to count number of digits in an integer.

Ans:-

#include<stdio.h>

int main()

{

int num,count=0;

printf("Enter num=");

scanf("%d",&num);

while(num!=0)

{

num/=10;

count++;

}

printf("Number of digits=%d",count);

return 0;

}

1. C program to reverse a number.

Ans:-

#include<stdio.h>

int main()

{

int num,rem=0,rev=0;

printf("Enter number=");

scanf("%d",&num);

while(num!=0)

{

rem=num%10;

rev=rev\*10+rem;

num=num/10;

}

printf("Reverse number is=%d",rev);

return 0;

}

1. C program to display factors of number.

Ans:-

#include<stdio.h>

int main()

{

int num,i;

printf("Enter numbr=");

scanf("%d",&num);

printf("Factors of %d are\n",num);

for(i=1;i<=num;i++)

{

if(num%i==0)

{

printf("%d\t",i);

}

}

return 0;

}

1. Check whether number is Armstrong, prime or palindrome using switch case.

Ans:-

#include<stdio.h>

int main()

{

int num,no;

int sum=0,rem=0,count=0,i,temp;

int ch;

CONTINUE: printf("1.Armstrong\n");

printf("2.Prime Number\n");

printf("3.Palindrome\n");

printf("Enter choice=");

scanf("%d",&ch);

switch(ch)

{

case 1 : printf("Enter number=");

scanf("%d",&num);

temp=num;

while(num!=0)

{

rem=num%10;

sum=sum+rem\*rem\*rem;

num=num/10;

}

printf("%d\n",sum);

if(sum==temp)

{

printf("%d is Armstrong number",temp);

}

else

{

printf("%d is not Armstrong number",temp);

}

break;

case 2: printf("Enter number=");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

if(num%i==0)

{

count++;

}

}

if(count==2)

{

printf("%d is prime number",num);

}

else

{

printf("%d is not prime number",num);

}

break;

case 3: printf("Enter num=");

scanf("%d",&num);

temp=num;

while(num!=0)

{

rem=num%10;

sum=sum\*10+rem;

num=num/10;

}

printf("%d\n",sum);

if(sum==temp)

{

printf("%d is palindrome",temp);

}

else

{

printf("%d is just a number",temp);

}

break;

default :printf("Please enter correct choice");

break;

}

}

1. Pattern –Half of \*.

Ans:-

#include<stdio.h>

int main()

{

int i,j,rows;

printf("Enter rows=");

scanf("%d",&rows);

for(i=1;i<=rows;i++)

{

for(j=1;j<=i;j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

1. Pattern –Half of numbers.

Ans:-

#include<stdio.h>

int main()

{

int i,j,rows;

printf("Enter rows=");

scanf("%d",&rows);

for(i=1;i<=rows;i++)

{

for(j=1;j<=i;j++)

{

printf("%d",j);

}

printf("\n");

}

return 0;

}

1. Pattern – Half of alphabets.

Ans:- A)

#include<stdio.h>

int main()

{

int i,j;

char alpha,ch='A';

printf("Enter last alphabet for pattern=");

scanf("%c",&alpha);

for(i=1;i<=(alpha-'A'+1);i++)

{

for(j=1;j<=i;j++)

{

printf("%c",ch);

}

ch++;

printf("\n");

}

return 0;

}

B)

#include<stdio.h>

int main()

{

int i,j;

char alpha,ch='A';

printf("Enter last alphabet for pattern=");

scanf("%c",&alpha);

for(i=1;i<=(alpha-'A'+1);i++)

{

for(j=1;j<=i;j++)

{

printf("%c",ch);

}

printf("\n");

}

return 0;

}

1. Pattern-Inverted of Half \*.

Ans:-

#include<stdio.h>

int main()

{

int i,j,rows;

printf("Enter rows=");

scanf("%d",&rows);

for(i=rows;i>=1;i--)

{

for(j=1;j<=i;j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

1. Pattern-Inverted of Half numbers.

Ans:-

#include<stdio.h>

int main()

{

int i,j,rows;

printf("Enter rows=");

scanf("%d",&rows);

for(i=rows;i>=1;i--)

{

for(j=1;j<=i;j++)

{

printf("%d",j);

}

printf("\n");

}

return 0;

}

1. Pattern-Inverted of Half of alphabets.

Ans:- A)

#include<stdio.h>

int main()

{

int i,j;

char alpha,ch='A';

printf("Enter last alphabet for pattern=");

scanf("%c",&alpha);

for(i=(alpha-'A'+1);i>=1;i--)

{

for(j=1;j<=i;j++)

{

printf("%c",ch);

}

ch++;

printf("\n");

}

return 0;

}

B)

#include<stdio.h>

int main()

{

int i,j;

char alpha,ch='A';

printf("Enter last alphabet for pattern=");

scanf("%c",&alpha);

for(i=(alpha-'A'+1);i>=1;i--)

{

for(j=1;j<=i;j++)

{

printf("%c",ch);

}

printf("\n");

}

return 0;

}

1. Pattern-full of \*.

Ans:-

#include <stdio.h>

int main()

{

int i, space, rows, k = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows;i++,k=0)

{

for (space = 1; space <= rows - i; space++)

{

printf(" ");

}

while(k != 2 \* i - 1)

{

printf("\* ");

k++;

}

printf("\n");

}

return 0;

}

1. Pattern-full of numbers.

Ans:- #include <stdio.h>

int main()

{

int i, space, rows, k = 0, count = 0, count1 = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; i++)

{

for (space = 1; space <= rows - i; ++space)

{

printf(" ");

count++;

}

while (k != 2 \* i - 1)

{

if (count <= rows - 1)

{

printf("%d ", i + k);

count++;

}

else

{

count1++;

printf("%d ", (i + k - 2 \* count1));

}

k++;

}

count1 = count = k = 0;

printf("\n");

}

return 0;

}

1. Pattern-Inverted full pyramid of \*.

Ans:-

#include <stdio.h>

int main()

{

int i, space, rows, k = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i>=1 ;i--,k=0)

{

for (space = 1; space <= rows - i; space++)

{

printf(" ");

}

while(k != 2 \* i - 1)

{

printf("\* ");

k++;

}

printf("\n");

}

return 0;

}

1. Floyd’s Triangle.

Ans:-A)

#include<stdio.h>

int main()

{

int i,j,rows,number=1;

printf("Enter rows=");

scanf("%d",&rows);

for(i=1;i<=rows;i++)

{

for(j=1;j<=i;j++)

{

printf("%d ",number);

number++;

}

printf("\n");

}

return 0;

}

B)

#include<stdio.h>

int main()

{

int i,j;

char alpha,ch='A';

printf("Enter last alphabet=");

scanf("%c",&alpha);

for(i=1;i<=(alpha-'A'+1);i++)

{

for(j=1;j<=i;j++)

{

printf("%c ",ch);

ch++;

}

printf("\n");

}

return 0;

}

1. Defining the single dimensional array.Access the member of array.

Ans:-

#include<stdio.h>

int main()

{

int arr[10];

int i;

printf("Enter array elements\n");

for(i=0;i<10;i++)

{

printf("Elements=");

scanf("%d",&arr[i]);

}

for(i=0;i<10;i++)

{

printf("a[%d]=%d\t",i,arr[i]);

}

return 0;

}

1. Defining double dimensional array.Access the member of array.

Ans:-

#include<stdio.h>

int main()

{

int arr[2][2],i,j;

printf("Array Elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr[i][j]);

}

}

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr[%d][%d]=%d\t",i,j,arr[i][j]);

}

}

return 0;

}

1. Find the Largest element in an array. Also display its index.

Ans:-

#include<stdio.h>

int main()

{

int arr[10],i,temp,index;

printf("Enter array Elements\n");

for(i=0;i<10;i++)

{

printf("Elements=");

scanf("%d",&arr[i]);

}

for(i=0;i<10;i++)

{

printf("a[%d]=%d\t",i,arr[i]);

}

printf("\n");

temp=arr[0];

for(i=0;i<10;i++)

{

if(arr[i]>temp)

{

temp=arr[i];

index=i;

}

}

printf("largest number of array is arr[%d]=%d",index,temp);

return 0;

}

1. Find the smallest element in an array. Also display its index.

Ans:-

#include<stdio.h>

int main()

{

int arr[10],i,temp,index;

printf("Enter array elements\n");

for(i=0;i<10;i++)

{

printf("Elements=");

scanf("%d",&arr[i]);

}

for(i=0;i<10;i++)

{

printf("arr[%d]=%d\t",i,arr[i]);

}

printf("\n");

temp=arr[0];

for(i=0;i<10;i++)

{

if(arr[i]<temp)

{

temp=arr[i];

index=i;

}

}

printf("Samllest number in an array is arr[%d]=%d",index,temp);

return 0;

}

1. Find Largest number in Matrix.

Ans:-

#include<stdio.h>

int main()

{

int arr[2][2],i,j,temp,index\_i,index\_j;

printf("Enter array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Eements=");

scanf("%d",&arr[i][j]);

}

}

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("%d\t",arr[i][j]);

}

}

printf("\n");

temp=arr[0][0];

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

if(arr[i][j]>temp)

{

temp=arr[i][j];

index\_i=i;

index\_j=j;

}

}

}

printf("Largest number of Matrix arr[%d][%d]=%d",index\_i,index\_j,temp);

return 0;

}

1. Find Smallest number in Matrix.

Ans:-

#include<stdio.h>

int main()

{

int arr[2][2],i,j,temp,index\_i,index\_j;

printf("Enter array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Eements=");

scanf("%d",&arr[i][j]);

}

}

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("%d\t",arr[i][j]);

}

}

printf("\n");

temp=arr[0][0];

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

if(arr[i][j]<temp)

{

temp=arr[i][j];

index\_i=i;

index\_j=j;

}

}

}

printf("Smallest number of Matrix arr[%d][%d]=%d",index\_i,index\_j,temp);

return 0;

}

1. Reverse the array.

Ans:-

#include<stdio.h>

int main()

{

int arr[10],i;

printf("Enter array elements\n");

for(i=0;i<10;i++)

{

printf("Elements=");

scanf("%d",&arr[i]);

}

for(i=0;i<10;i++)

{

printf("arr[%d]=%d\t",i,arr[i]);

}

printf("\n");

printf("Reverse array is\n");

for(i=9;i>=0;i--)

{

printf("arr[%d]=%d\t",i,arr[i]);

}

return 0;

}

1. Add single dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[5],arr2[5],arr3[5],i;

printf("Enter 1st array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr1[i]);

}

printf("Enter 2nd array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr2[i]);

}

printf("1st array elements\n");

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

{

printf("arr1[%d]=%d\t",i,arr1[i]);

}

printf("\n");

printf("2nd array elements\n");

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

{

printf("arr2[%d]=%d\t",i,arr2[i]);

}

printf("\n");

printf("Addition of elements\n");

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

{

arr3[i]=arr1[i] + arr2[i];

printf("arr3[%d]=%d\t",i,arr3[i]);

}

return 0;

}

1. Subtarct single dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[5],arr2[5],arr3[5],i;

printf("Enter 1st array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr1[i]);

}

printf("Enter 2nd array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr2[i]);

}

printf("1st array elements\n");

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

{

printf("arr1[%d]=%d\t",i,arr1[i]);

}

printf("\n");

printf("2nd array elements\n");

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

{

printf("arr2[%d]=%d\t",i,arr2[i]);

}

printf("\n");

printf("Subtarction of elements\n");

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

{

arr3[i]=arr1[i] - arr2[i];

printf("arr3[%d]=%d\t",i,arr3[i]);

}

return 0;

}

1. Multiply single dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[5],arr2[5],arr3[5],i;

printf("Enter 1st array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr1[i]);

}

printf("Enter 2nd array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr2[i]);

}

printf("1st array elements\n");

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

{

printf("arr1[%d]=%d\t",i,arr1[i]);

}

printf("\n");

printf("2nd array elements\n");

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

{

printf("arr2[%d]=%d\t",i,arr2[i]);

}

printf("\n");

printf("Multiplication of elements\n");

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

{

arr3[i]=arr1[i] \* arr2[i];

printf("arr3[%d]=%d\t",i,arr3[i]);

}

return 0;

}

1. Divide operation on single array

Ans:-

#include<stdio.h>

int main()

{

int arr1[5],arr2[5],arr3[5],i;

printf("Enter 1st array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr1[i]);

}

printf("Enter 2nd array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr2[i]);

}

printf("1st array elements\n");

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

{

printf("arr1[%d]=%d\t",i,arr1[i]);

}

printf("\n");

printf("2nd array elements\n");

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

{

printf("arr2[%d]=%d\t",i,arr2[i]);

}

printf("\n");

printf("Division of elements\n");

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

{

arr3[i]=arr1[i] / arr2[i];

printf("arr3[%d]=%d\t",i,arr3[i]);

}

return 0;

}

1. Modulus operation of array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[5],arr2[5],arr3[5],i;

printf("Enter 1st array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr1[i]);

}

printf("Enter 2nd array elements\n");

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

{

printf("Elements=");

scanf("%d",&arr2[i]);

}

printf("1st array elements\n");

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

{

printf("arr1[%d]=%d\t",i,arr1[i]);

}

printf("\n");

printf("2nd array elements\n");

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

{

printf("arr2[%d]=%d\t",i,arr2[i]);

}

printf("\n");

printf("Modulus of elements of an arry\n");

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

{

arr3[i]=arr1[i] % arr2[i];

printf("arr3[%d]=%d\t",i,arr3[i]);

}

return 0;

}

1. Add double dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[2][2],arr2[2][2],arr3[2][2];

int i,j;

printf("Enter 1st array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Enter 2nd array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr2[i][j]);

}

}

printf("\n");

printf("1st Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("2nd Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr2[%d][%d]=%d\t",i,j,arr2[i][j]);

}

}

printf("\n");

printf("Addition of Matrix\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

arr3[i][j]=arr1[i][j]+arr2[i][j];

printf("arr3[%d][%d]=%d\t",i,j,arr3[i][j]);

}

}

return 0;

}

1. Subtract double dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[2][2],arr2[2][2],arr3[2][2];

int i,j;

printf("Enter 1st array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Enter 2nd array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr2[i][j]);

}

}

printf("\n");

printf("1st Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("2nd Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr2[%d][%d]=%d\t",i,j,arr2[i][j]);

}

}

printf("\n");

printf("Subtraction of Matrix\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

arr3[i][j]=arr1[i][j]-arr2[i][j];

printf("arr3[%d][%d]=%d\t",i,j,arr3[i][j]);

}

}

return 0;

}

1. Multiply double dimensional array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[2][2],arr2[2][2],arr3[2][2];

int i,j;

printf("Enter 1st array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Enter 2nd array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr2[i][j]);

}

}

printf("\n");

printf("1st Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("2nd Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr2[%d][%d]=%d\t",i,j,arr2[i][j]);

}

}

printf("\n");

printf("Multiplication of Matrix\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

arr3[i][j]=arr1[i][j]\*arr2[i][j];

printf("arr3[%d][%d]=%d\t",i,j,arr3[i][j]);

}

}

return 0;

}

1. Division of double dimension array.

Ans:-

#include<stdio.h>

int main()

{

int arr1[2][2],arr2[2][2],arr3[2][2];

int i,j;

printf("Enter 1st array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Enter 2nd array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr2[i][j]);

}

}

printf("\n");

printf("1st Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("2nd Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr2[%d][%d]=%d\t",i,j,arr2[i][j]);

}

}

printf("\n");

printf("Division of Matrix\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

arr3[i][j]=arr1[i][j]/arr2[i][j];

printf("arr3[%d][%d]=%d\t",i,j,arr3[i][j]);

}

}

return 0;

}

1. Modulus of Matrix.

Ans:-

#include<stdio.h>

int main()

{

int arr1[2][2],arr2[2][2],arr3[2][2];

int i,j;

printf("Enter 1st array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Enter 2nd array elements\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Elements=");

scanf("%d",&arr2[i][j]);

}

}

printf("\n");

printf("1st Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("2nd Matrix elements\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

printf("arr2[%d][%d]=%d\t",i,j,arr2[i][j]);

}

}

printf("\n");

printf("Modulus of Matrix\n");

for(i=0;i<2;i++)

{

printf("\n");

for(j=0;j<2;j++)

{

arr3[i][j]=arr1[i][j]%arr2[i][j];

printf("arr3[%d][%d]=%d\t",i,j,arr3[i][j]);

}

}

return 0;

}

1. C program to find transpose of Matrix.

Ans:-

#include<stdio.h>

int main()

{

int row,col;

printf("Enter row size=");

scanf("%d",&row);

printf("Enter column size=");

scanf("%d",&col);

int arr1[row][col],t\_arr[col][row];

int i,j;

printf("Enter array elements\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

printf("Elements=");

scanf("%d",&arr1[i][j]);

}

}

printf("Matrix elements are\n");

for(i=0;i<row;i++)

{

printf("\n");

for(j=0;j<col;j++)

{

printf("arr1[%d][%d]=%d\t",i,j,arr1[i][j]);

}

}

printf("\n");

printf("Transpose of Matrix\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

t\_arr[j][i]=arr1[i][j];

}

}

for(i=0;i<col;i++)

{

for(j=0;j<row;j++)

{

printf("t\_arr[%d][%d]=%d\t",i,j,t\_arr[i][j]);

}

printf("\n");

}

return 0;

}

1. W.A.P. to access elements of array using pointers.

Ans:-

#include<stdio.h>

int main()

{

int \*parr,i;

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

{

printf("Enter array elements=");

scanf("%d",&parr[i]);

}

printf("Array Elements=");

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

{

printf("parr[%d]=%d\t",i,parr[i]);

}

return 0;

}

1. W.A.P. to access character members of array using pointers.

Ans:-

#include<stdio.h>

int main()

{

char \*ptr[5]={"Prasad","Roopa","Supriya","Nagarjuna","Sai"};

int i;

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

{

printf("ptr[%d]=%s\n",i,ptr[i]);

}

return 0;

}

1. W.A.P. to access matrix elements using double pointer.

Ans:-

#include<stdio.h>

int main()

{

int row,col;

printf("Enter row size=");

scanf("%d",&row);

printf("Enter column size=");

scanf("%d",&col);

int arr[row][col],i,j,row\_n,col\_n;

printf("Enter Matrix Elements:-\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

printf("Elements:-");

scanf("%d",&arr[i][j]);

}

}

printf("Matrix Elements are:-\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

printf("%d\t",arr[i][j]);

}

printf("\n");

}

printf("\n");

printf("\n");

/\*Accessing Specific member of specific row & column\*/

printf("Access Elements of Matrix:\n");

printf("Enter row=");

scanf("%d",&row\_n);

printf("Enter col=");

scanf("%d",&col\_n);

printf("%d",\*(\*(arr+row\_n)+col\_n));

/\*Accessing all members\*/

printf("Access Elements of Matrix:\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

printf("%d\t",\*(\*(arr+i)+j));

}

printf("\n");

}

return 0;

}

1. W.A.P to represent pointer representation.

Ans:-

#include<stdio.h>

int main()

{

int a=10;

int \*ptr;

int \*\*pa;

ptr=&a;

pa=&ptr;

printf("%d\n",a);

printf("%d\n",\*ptr);

printf("%d\n",ptr);

printf("%d\n",pa);

printf("%d\n",\*pa);

printf("%d\n",\*\*pa);

/\*Arithmetic of Pointers\*/

/\*Note 1:- We can’t perform arithmetic operation on two pointer i.e. (ptr1+ptr2) or (ptr1-ptr2) is invalid

Note 2:- We can’t do Multiplication, Division or Modulus operation on pointer i.e. (ptr\*2) or (ptr/2) or(ptr%2)\*/

printf("\n");

printf("\n");

x=ptr+1;

y=ptr-2;

printf("%d\n",x);

printf("%d\n",y);

return 0;

}

1. W.A.P. for simple function implementation.

Ans:-

/\*Simple Function\*/

#include<stdio.h>

void add()

{

int a,b,c;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

c=a+b;

printf("Addition of two number=%d",c);

}

int main()

{

add();

return 0;

}

1. W.A.P. to implement function using call by value.

Ans:-

#include<stdio.h>

void interchange (int x,int y);

void interchange (int x,int y)

{

int temp;

temp=x;

x=y;

y=temp;

printf("After interchange value\n");

printf("m=%d\n",x);

printf("n=%d\n",y);

}

int main()

{

int m,n;

printf("Enter m=");

scanf("%d",&m);

printf("Enter n=");

scanf("%d",&n);

printf("Before interhcange value\n");

printf("m=%d\n",m);

printf("n=%d\n",n);

interchange(m,n);

return 0;

}

1. W.A.P. to implement function using call by reference.

Ans:-

#include<stdio.h>

void interchange (int \*x,int \*y);

void interchange (int \*x,int \*y)

{

int temp;

temp=\*x;

\*x=\*y;

\*y=temp;

printf("After interchange value\n");

printf("m=%d\n",\*x);

printf("n=%d\n",\*y);

}

int main()

{

int m,n;

printf("Enter m=");

scanf("%d",&m);

printf("Enter n=");

scanf("%d",&n);

printf("Before interhcange value\n");

printf("m=%d\n",m);

printf("n=%d\n",n);

interchange(&m,&n);

return 0;

}

1. W.A.P. to implement fucnton using Recursion.

Ans:-

#include<stdio.h>

int fact(int n)

{

if(n==0)

{

return 1;

}

else

{

return (n\*fact(n-1));

}

}

int main()

{

int num,val,result;

printf("Enter num=");

scanf("%d",&num);

val=num;

result=fact(num);

printf("Factorial of %d=%d",val,result);

return 0;

}

1. W.A.P. to implement function for “No Argument & No Return”

Ans:-

#include<stdio.h>

void even\_or\_odd(void);

void even\_or\_odd()

{

int num;

printf("Enter num=");

scanf("%d",&num);

if(num%2==0)

{

printf("%d is even\n",num);

}

else

{

printf("%d is odd\n",num);

}

}

int main()

{

even\_or\_odd();

return 0;

}

1. W.A.P. to implement function using “wih argument & no return”

Ans:-

#include<stdio.h>

void even\_or\_odd(int num);

void even\_or\_odd(int num)

{

if(num%2==0)

{

printf("%d is even\n",num);

}

else

{

printf("%d is odd\n",num);

}

}

int main()

{

int val;

printf("Enter value=");

scanf("%d",&val);

even\_or\_odd(val);

return 0;

}

1. W.A.P. to implement function for “No Argument Yes Return”

Ans:-

#include<stdio.h>

int even\_or\_odd(void);

int even\_or\_odd()

{

int num,result;

printf("Enter number=");

scanf("%d",&num);

result=num%2;

return result;

}

int main()

{

int val;

val=even\_or\_odd();

if(val==0)

{

printf("Number is even\n");

}

else

{

printf("Number is odd\n");

}

return 0;

}

1. W.A.P. to implement function “with argument & return”

Ans:-

#include<stdio.h>

int even\_or\_odd(int num);

int even\_or\_odd(int num)

{

int result;

result=num%2;

return result;

}

int main()

{

int num,val;

printf("Enter number=");

scanf("%d",&num);

val=even\_or\_odd(num);

if(val==0)

{

printf("%d is even\n",num);

}

else

{

printf("%d is odd\n",num);

}

return 0;

}

1. W.A.P. to find multiplication of two number using bitwise operator.

Ans:-

#include<stdio.h>

int main()

{

int a,b,x,y,result=0;

printf("Enter a=");

scanf("%d",&a);

printf("Enter b=");

scanf("%d",&b);

x=a;

y=b;

while(b>0)

{

if(b & 1==1)

{

result=result+a;

}

a=a<<1;

b=b>>1;

}

printf("Multiplication result is given as:\n");

printf("%d \* %d = %d",x,y,result);

}

1. W.A.P. to find number is even or odd using arithmetic ,shift operator & bitwise operator.

Ans:- #include<stdio.h>

int main()

{

int num;

printf("Enter number=");

scanf("%d",&num);

/\*Using '%' Operator\*/

if(num%2==0)

{

printf("%d is Even Number",num);

}

else

{

printf("%d is Odd Number",num);

}

/\*Uisng '/' Opeartor\*/

if((num/2)\*2==num)

{

printf("%d is Even Number",num);

}

else

{

printf("%d is Odd Number",num);

}

/\*Using Shift Operator\*/

if((num>>1)<<1 == num)

{

printf("%d is Even Number",num);

}

else

{

printf("%d is Odd Number",num);

}

/\*Using Bitwise Operator\*/

if((num&1)==1)

{

printf("%d is Odd Number",num);

}

else

{

printf("%d is Even Number",num);

}

return 0;

}

1. W.A.P. to printf a number upto N without semicolon

Ans:-

#include<stdio.h>

#define N 50

int main(int num)

{

if (num <= N && printf("%d ", num) && main(num + 1))

{

}

}

1. W.A.P. to print statement without semicolon

Ans:-

#include<stdio.h>

int main()

{

if(printf("Mobiveil"))

{

}

return 0;

}

1. W.A.P. to find G.C.D. of two number

Ans:-

#include <stdio.h>

int main()

{

int n1, n2, i, gcd;

printf("Enter 1st Number=");

scanf("%d",&n1);

printf("Enter 2nd Number=");

scanf("%d",&n2);

for(i=1; i <= n1 && i <= n2; ++i)

{

if(n1%i==0 && n2%i==0)

{

gcd = i;

}

}

printf("G.C.D of %d and %d is %d", n1, n2, gcd);

return 0;

}

1. W.A.P. to find L.C.M. of two numbers.

Ans:-

#include<stdio.h>

int main()

{

int num1,num2,max,i;

printf("Enter num1=");

scanf("%d",&num1);

printf("Enter num2=");

scanf("%d",&num2);

max=(num1>num2) ? num1 : num2;

printf("%d\n",max);

while(1)

{

if(max%num1==0 && max%num2==0)

{

printf("L.C.M. of %d & %d =%d\n",num1,num2,max);

break;

}

max++;

}

return 0;

}

1. W.A.P. to find system is little or big endian

Ans:-

#include <stdio.h>

int main()

{

unsigned int i = 1;

char \*c = (char\*)&i;

if (\*c)

printf("Little endian");

else

printf("Big endian");

getchar();

return 0;

}

1. W.A.P. to convert little to big endian

Ans:-

#include <stdio.h>

int changed\_endian(int num)

{

int byte0, byte1, byte2, byte3;

byte0 = (num & 0x000000FF) >> 0 ;

byte1 = (num & 0x0000FF00) >> 8 ;

byte2 = (num & 0x00FF0000) >> 16 ;

byte3 = (num & 0xFF000000) >> 24 ;

return((byte0 << 24) | (byte1 << 16) | (byte2 << 8) | (byte3 << 0));

}

int main()

{

unsigned int number=1;

unsigned int new\_number;

new\_number=changed\_endian(number);

char \*c=(char\*)&new\_number;

printf("%d\n",number);

printf("%d\n",new\_number);

if(\*c)

{

printf("System is little endian");

}

else

{

printf("System is big endian");

}

return 0;

}