Overlapadd.C

#include<stdio.h>

int cir\_out[20][10];

void circonv(int x[],int h[], int len, int arr\_flag){

int i,k,modval;

int maxlen = len;

int y[maxlen];

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

{

y[i]=0;

for(k=0;k<maxlen;k++)

{

if((i-k)<0)

{

modval = maxlen+(i-k);

y[i] += (x[k]\*h[modval]);

}

else

{

y[i] += (x[k]\*h[(i-k)]);

}

}

cir\_out[arr\_flag][i] = y[i];

}

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

{

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

}

printf("\n");

}

void overlapadd(int l, int n, int len, int out\_len)

{

int i,j;

int overlap\_arr[len][out\_len];

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

{

for(j=0; j<out\_len ; j++)

{

overlap\_arr[i][j] = 0;

}

}

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

{

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

{

if(j+(l\*i) < out\_len)

{

overlap\_arr[1][j+(l\*i)] += cir\_out[i][j];

printf("arr:%d,\t i:%d,\t j:%d\n", overlap\_arr[1][j+(l\*i)],i,j+(l\*i));

}

else

break;

}

}

for(j=0; j<out\_len ; j++)

{

printf("%d\t", overlap\_arr[1][j]);

}

}

int main(){

int h[20], x[20];

int lx,lh,i;

printf("Enter length of x[n]\n");

scanf("%d",&lx);

printf("Enter length of h[n]\n");

scanf("%d",&lh);

printf("Enter elements for x[n]\n");

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

{

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

}

printf("Enter elements for h[n]\n");

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

{

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

}

int l = 4 , m = lh, n = l+m-1;

int temp[n], tempFlag=1, overlap\_arr\_length=0, count = 0;

if(lh < n)

{

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

h[i] = 0;

}

while(tempFlag != 0)

{

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

{

if(i < l && count < lx)

{

temp[i] = x[count];

count++;

}

else

temp[i] = 0;

}

circonv(temp, h, n, tempFlag);

tempFlag++;

if(count == lx)

{

overlap\_arr\_length = tempFlag;

tempFlag = 0;

}

}

overlapadd(l,n, overlap\_arr\_length, (lx+lh-1));

}

/\* OUTPUT:

Enter length of x[n]

13

Enter length of h[n]

4

Enter elements for x[n]

1 2 -1 3 -2 1 0 3 4 -2 3 1 4

Enter elements for h[n]

1 -1 0 1

Linear convolution by overlapadd

1 1 -3 5 -3 2 2 1 2 -6 8 2 -2 2 1 1

\*/