

1. Delay, Advance, Attenuation and Amplification operations on Signal

Code:

```
#include<stdio.h>
#include<math.h>

int arrow,N,k;
static int org_input[100],sample[100];

int main(void)
{
    int i,j,c,l;
    printf("*****\n\n");
    printf("\t\t Operations on Signals\n\n");
    printf("*****\n\n");
    do
    {
        printf("\n\nOperations On Signals :-\n");
        printf("***** Transformation of In-Dependant Variable *****\n");
        printf("1. Delay\n");
        printf("2. Advance\n");
        printf("3. Fold\n");
        printf("4. Down Sampling (Compression)\n");
        printf("5. Up Sampling(Expansion)\n");
        printf("***** Transformation of Dependant Variable *****\n");
        printf("6. Amplification\n");
        printf("7. De-Amplification\n");
        printf("8. Exit\n");

        printf("\n*****\n");
        printf("Enter your Choice :: ");
        scanf("%d",&c);

        switch(c)
        {
            case 1:
                //Delay Operation
                printf("\nYou Selected option: 1. Delay\n");
                printf("\nEnter Number of Samples N:");
                scanf("%d",&N);
                printf("Enter Samples :\n");
                for(i=0;i<N;i++)
                {
                    printf("x(%d) :",i);
                    scanf("%d",&org_input[i]);
                }
                printf("\nEnter the arrow Position :");
                scanf("%d",&arrow);
                printf("\nEnter Delay by (k) :");
                scanf("%d",&k);
```

```

for(j=1;j<k;j++)
{
    sample[j]= 0 ;
}
l=0;
for(i=0;i<N;i++)
{
    sample[k+i]=org_input[l];
    l++;
}
printf("Input Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",org_input[i]);
}
printf("\n");
printf("Delayed Signal is :\n");
printf("{\t");
for(i=0;i<(N+k);i++)
{
    printf("%d\t",sample[i]);
}
printf("}");
printf("\nArrow Position at %d and sample at position is %d",arrow,sample[arrow]);
getch();
break;

```

case 2:

//Advance

```

printf("\nYou Selected option: 2. Advance\n");
printf("\nEnter Number of Samples N:");
scanf("%d",&N);
printf("Enter Samples :\n");
for(i=0;i<N;i++)
{
    printf("x(%d) :",i);
    scanf("%d",&org_input[i]);
}
printf("\nEnter the arrow Position :");
scanf("%d",&arrow);
printf("\nEnter Advance by (k) :");
scanf("%d",&k);

l=0;
for(i=0;i<N;i++)

```

```

{
    sample[i]=org_input[i];
}
for(j=0;j<k;j++)
{
    sample[N+j]= 0 ;
}
printf("Input Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",org_input[i]);
}
printf("\n");
printf("Advance Signal is :\n");
printf("{\t");
for(i=0;i<(N+k);i++)
{
    printf("%d\t",sample[i]);
}
printf("}");
printf("\nArrow Position at %d and sample at position is %d",arrow+k,sample[arrow+k]);
getch();
break;

```

case 3:

//Folding

```

printf("\nYou Selected option: 3. Fold\n");
printf("\nEnter Number of Samples N:");
scanf("%d",&N);
printf("Enter Samples :\n");
for(i=0;i<N;i++)
{
    printf("x(%d) :",i);
    scanf("%d",&org_input[i]);
}
printf("\nEnter the arrow Position :");
scanf("%d",&arrow);

k = N-1;
for(i=0;i<N;i++)
{
    sample[i] = org_input[k];
    k--;
}
printf("Input Signal is %d:\n",k);
printf("{\t");
for(i=0;i<N;i++)
{

```

```

        printf("%d\t",org_input[i]);
    }
    printf("\n");
    printf("Fold Signal is :\n");
    printf("{\t");
    for(i=0;i<N;i++)
    {
        printf("%d\t",sample[i]);
    }
    printf("}");
    sample[arrow] = org_input[arrow];
    printf("\nArrow Position at %d and sample at position is %d",arrow,sample[arrow]);
    getch();

    break;

```

```

case 4:
//Compression
    printf("\nYou Selected option: 4. Down Sampling (Compression)\n");
    printf("\nEnter Number of Samples N:");
    scanf("%d",&N);
    printf("Enter Samples :\n");
    for(i=0;i<N;i++)
    {
        printf("x(%d) :",i);
        scanf("%d",&org_input[i]);
    }
    printf("\nEnter the arrow Position :");
    scanf("%d",&arrow);
    printf("\nEnter Compression by (u) :");
    scanf("%d",&k);

    printf("Input Signal is :\n");
    printf("{\t");
    for(i=0;i<N;i++)
    {
        printf("%d\t",org_input[i]);
    }
    printf("\n");
    printf("Compressed Signal is :-\n");
    printf("{\t");
    for(i=0;i<N;i++)
    {
        printf("%d\t",org_input[i*k]);
    }
    printf("\n");
    printf("\nArrow Position at %d and sample at position is %d",arrow,org_input[arrow]);
    getch();

    break;

```

case 5:

//Expansion

```

printf("\nYou Selected option: 5. Up Sampling (Expansion)\n");
printf("\nEnter Number of Samples N:");
scanf("%d",&N);
printf("Enter Samples :\n");
for(i=0;i<N;i++)
{
    printf("x(%d) :",i);
    scanf("%d",&org_input[i]);
}
printf("\nEnter the arrow Position :");
scanf("%d",&arrow);
printf("\nEnter Expansion by (1/u) :");
scanf("%d",&k);

printf("Input Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",org_input[i]);
}
printf("\n");
printf("Expansion Signal is :-\n");
printf("{\t");
for(i=0;i<N*k;i++)
{
    printf("%d\t",org_input[i/k]);
}
printf("\n");
printf("\nArrow Position at %d and sample at position is %d",arrow,org_input[arrow]);
getch();
break;

```

case 6:

//Amplification

```

printf("\nYou Selected option: 6. Amplification\n");
printf("\nEnter Number of Samples N:");
scanf("%d",&N);
printf("Enter Samples :\n");
for(i=0;i<N;i++)
{
    printf("x(%d) :",i);
    scanf("%d",&org_input[i]);
}
printf("\nEnter the arrow Position :");
scanf("%d",&arrow);
printf("\nEnter Amplification by (A) :");
scanf("%d",&k);

```

```

printf("Input Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",org_input[i]);
}
printf("}\n");
l=0;
for(j=0;j<N;j++)
{
    l = org_input[j];
    l = l*k;
    sample[j]=l;
}
printf("Amplified Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",sample[i]);
}
printf("}\n");
printf("\nArrow Position at %d and sample at position is %d",arrow,sample[arrow]);
getch();
break;

```

case 7:

//De-Amplification

```

printf("\nYou Selected option: 7. De-Amplification\n");
printf("\nEnter Number of Samples N:");
scanf("%d",&N);
printf("Enter Samples :\n");
for(i=0;i<N;i++)
{
    printf("x(%d) :",i);
    scanf("%d",&org_input[i]);
}
printf("\nEnter the arrow Position :");
scanf("%d",&arrow);
printf("\nEnter De-Amplification by (1/A) :");
scanf("%d",&k);

printf("Input Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",org_input[i]);
}
printf("}\n");
l=0;
for(j=0;j<N;j++)

```

```

{
    l = org_input[j];
    l = l*1/k;
    sample[j]=l;
}
printf("De-Amplified Signal is :\n");
printf("{\t");
for(i=0;i<N;i++)
{
    printf("%d\t",sample[i]);
}
printf("\n");
printf("\nArrow Position at %d and sample at position is %d",arrow,sample[arrow]);
getch();
break;

case 8:
    printf("\nEnded Successfully !!");
    break;

default:
    printf("Incorrect option.....");
}
printf("\n");
}while(c != 8);
printf("\n\n");
}

```

Output:

```
"F:\C++\operations on Signal.exe"
*****
Operations on Signals
*****

Operations On Signals :-
***** Transformation of In-Dependant Variable *****
1. Delay
2. Advance
3. Fold
4. Down Sampling (Compression)
5. Up Sampling(Expansion)
***** Transformation of Dependant Variable *****
6. Amplification
7. De-Amplification
8. Exit

*****
Enter your Choice ::
```


Delay Operation :

```

*****
Enter your Choice :: 1

You Selected option: 1. Delay

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0

Enter Delay by (k) :2
Input Signal is :
{      1      3      2      5      }
Delayed Signal is :
{      0      0      1      3      2      5      }
Arrow Position at 0 and sample at position is 0

```

Advance Operation :

```

*****
Enter your Choice :: 2

You Selected option: 2. Advance

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0

Enter Advance by (k) :2
Input Signal is :
{      1      3      2      5      }
Advance Signal is :
{      1      3      2      5      0      0      }
Arrow Position at 2 and sample at position is 2_

```

Fold Operation :

```

Enter your Choice :: 3
You Selected option: 3. Fold

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0
Input Signal is -1:
{      1      3      2      5      }
Fold Signal is :
{      5      2      3      1      }
Arrow Position at 0 and sample at position is 1

```

Down Sampling Operation :

```

Enter your Choice :: 4
You Selected option: 4. Down Sampling (Compression)

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0
Enter Compression by (u) :2
Input Signal is :
{      1      3      2      5      }
Compressed Signal is :-
{      1      2      0      0      }

Arrow Position at 0 and sample at position is 1

```

Up Sampling :

```

Enter your Choice :: 5
You Selected option: 5. Up Sampling (Expansion)
Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0
Enter Expansion by (1/u) :2
Input Signal is :
{      1      3      2      5      }
Expansion Signal is :-
{      1      1      3      3      2      2      5      5      }

Arrow Position at 0 and sample at position is 1

```

Amplification Operation :

```

Enter your Choice :: 6
You Selected option: 6. Amplification
Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0
Enter Amplification by (A) :2
Input Signal is :
{      1      3      2      5      }
Amplified Signal is :
{      2      6      4      10      }

Arrow Position at 0 and sample at position is 2

```

De – Amplification :

```
Enter your Choice :: 7
You Selected option: 7. De-Amplification

Enter Number of Samples N:4
Enter Samples :
x(0) :1
x(1) :3
x(2) :2
x(3) :5

Enter the arrow Position :0

Enter De-Amplification by (1/A) :2
Input Signal is :
{      1.00      3.00      2.00      5.00      }
De-Amplified Signal is :
{      0.50      1.50      1.00      2.50      }

Arrow Position at 0 and sample at position is 0.50
```