DSIP

Experiment 1

Part A

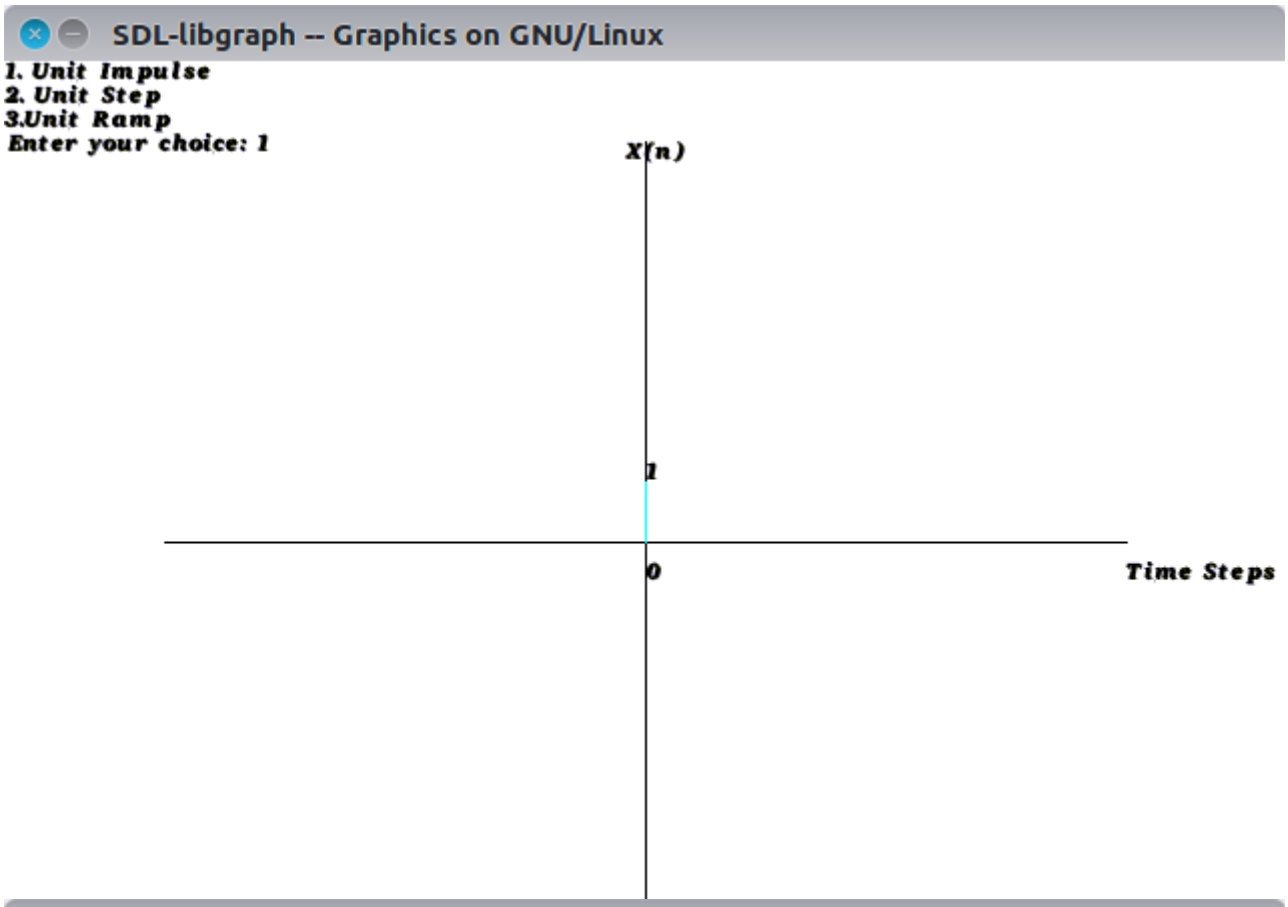
Code:

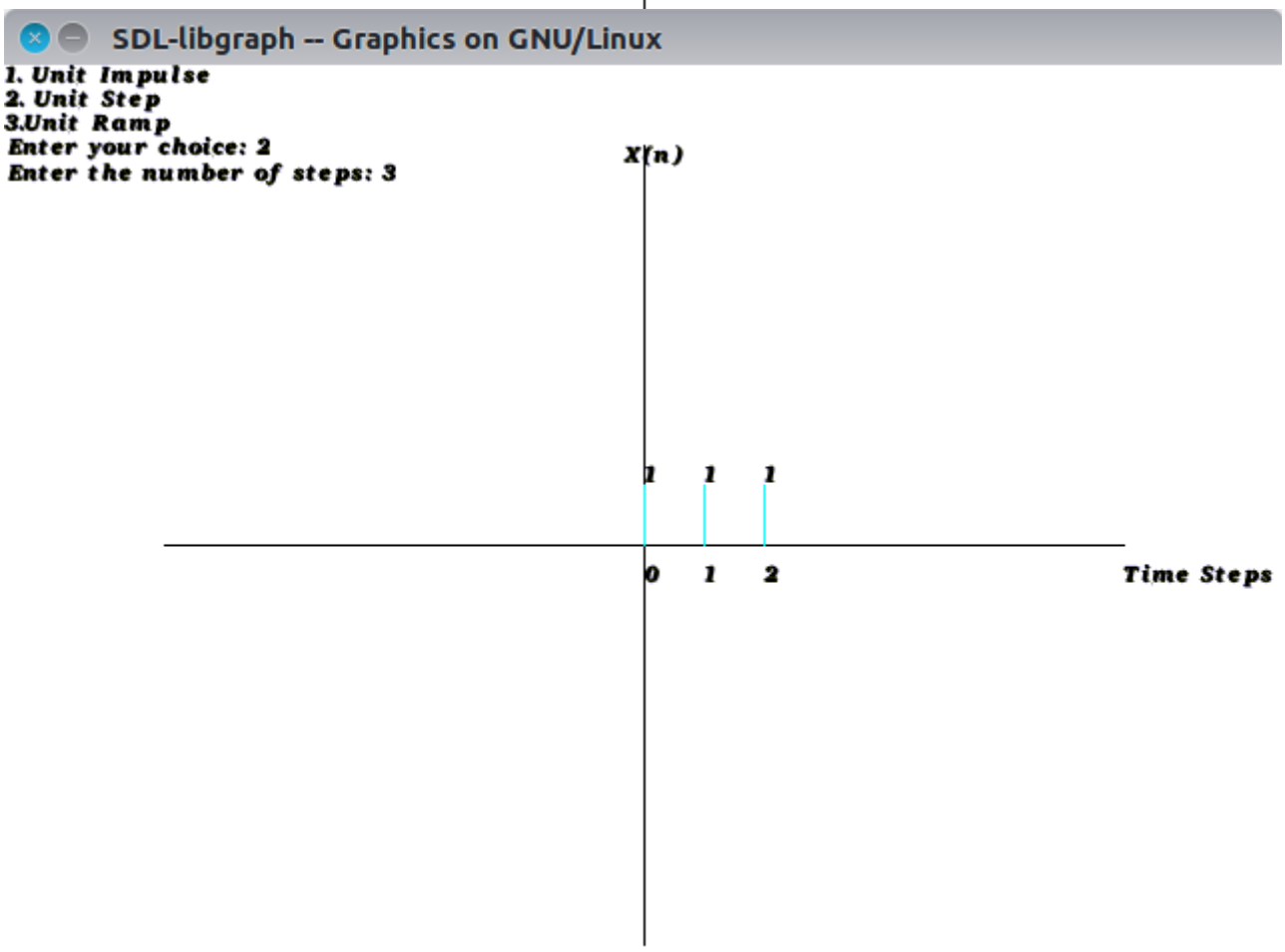
```
#include <graphics.h>
//#include <conio.h>
#include<X11/Xlib.h>
#define origX 320
#define origY 240

void axis(int type, int x, int y, int length);
void impulse(int x, int val);
void unit_ramp(int steps);
void unit_step(int steps);

void wait_for_char()
{

   //Wait for a key press
   int in = 0;

   while (in == 0) {
      in = getchar();
   }
}

int main() {
        XInitThreads();
        int gd = DETECT, gm;
        int opt;
        initgraph(&gd, &gm, NULL);
        // Drawing X-axis
        axis(0, origX, origY, 240);
        // Drawing Y-axis
        axis(1, origX, origY, 200);
        printf("1. Unit Impulse\n2. Unit Step\n3.Unit Ramp\nEnter your choice: ");
        scanf("%d", &opt);
        switch(opt) {
                case 1:
                        impulse(0, 1);
                        break;
                case 2:
                        printf("Enter the number of steps: ");
                        scanf("%d", &opt);
                        unit_step(opt);
                        break;
                case 3:
```

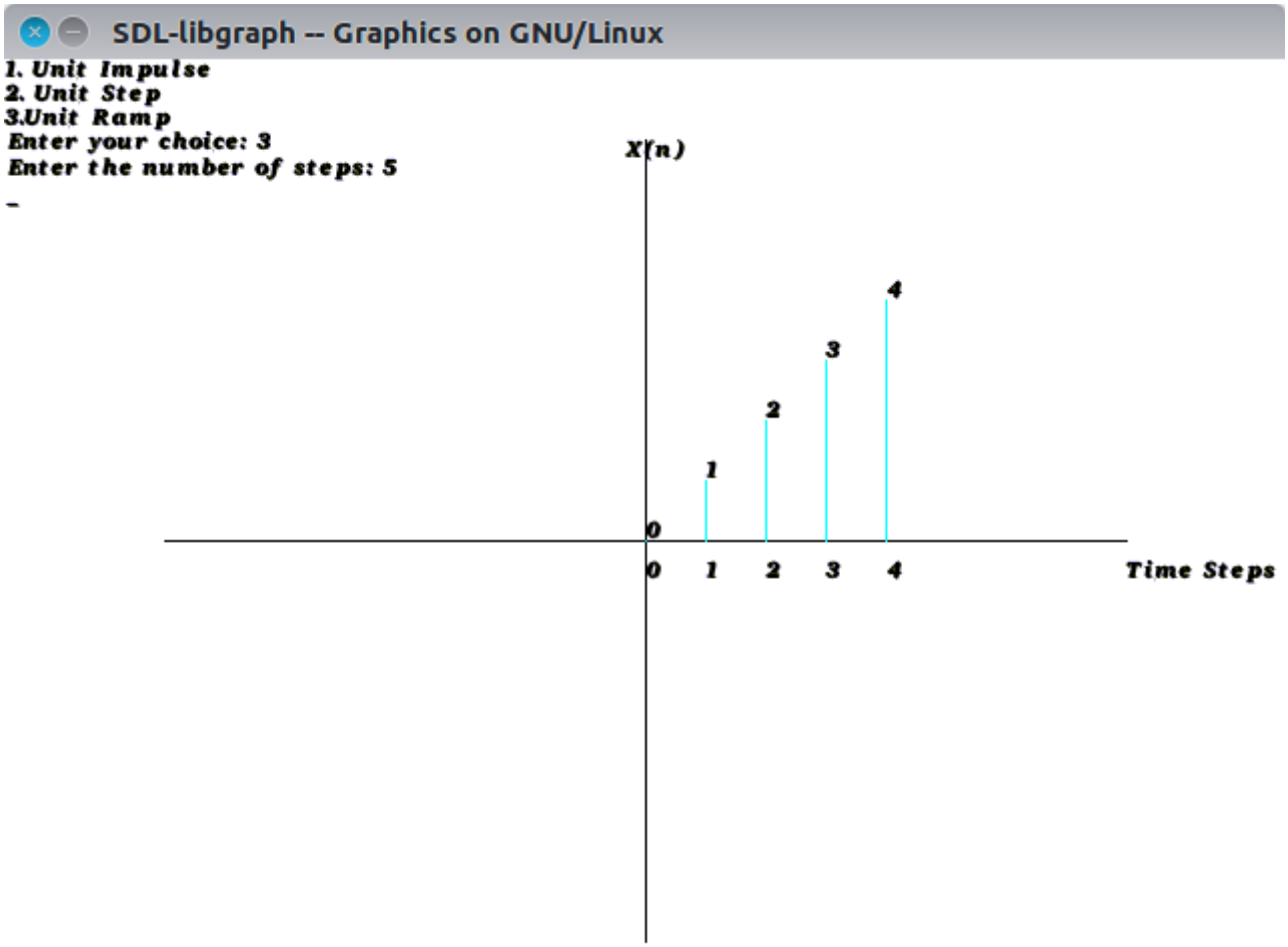
```
                printf("Enter the number of steps: ");
                scanf("%d", &opt);
                unit_ramp(opt);
                break;
        default:
                printf("Invalid Choice");
                break;
    }
    //getch();
    //delay(5000);
    //scanf("%d", &opt);
    wait_for_char();
    closegraph();
    return 0;
}

void axis(int type, int x, int y, int length) {
    int i;
    if (type == 0) {
        line(x - length, y, x + length, y);
        outtextxy(x + length, y + 10, "Time Steps");
    }
    else {
        line(x, y - length, x, y + length);
        outtextxy(x - 10, y - length, "X(n)");
    }
}

void impulse(int x, int val) {
    int new_x = origX + 30*x;
    int new_y = origY - 30*val;
    char x_label[2], y_label[2];
    y_label[0] = (char) (val + 48);
    y_label[1] = '\0';
    x_label[0] = (char)(x+48);
    x_label[1] = '\0';
    setcolor(RED);
    line(new_x, origY, new_x, new_y);
    outtextxy(new_x, new_y - 10, y_label);
    outtextxy(new_x, origY + 10, x_label);
    setcolor(WHITE);
}

void unit_ramp(int steps) {
    int i;
    for (i = 0; i < steps; i++)
        impulse(i, i);
}

void unit_step(int steps) {
    int i;
    for (i = 0; i < steps; i++)
```

```
                impulse(i, 1);
}
```

Output:







Part B

Code:

```
#include <graphics.h>
#include <conio.h>
#include <stdio.h>
#define origX 320
#define origY 240

void axis(int type, int x, int y, int length) {
	int i;
	if (type == 0) {
		line(x - length, y, x + length, y);
		outtextxy(x + length, y + 10, "Time Steps");
	}
	else {
		line(x, y - length, x, y + length);
		outtextxy(x - 10, y - length, "X(n)");
	}
}

void impulse(int x, int val) {
	int new_x = origX + 30*x;
```

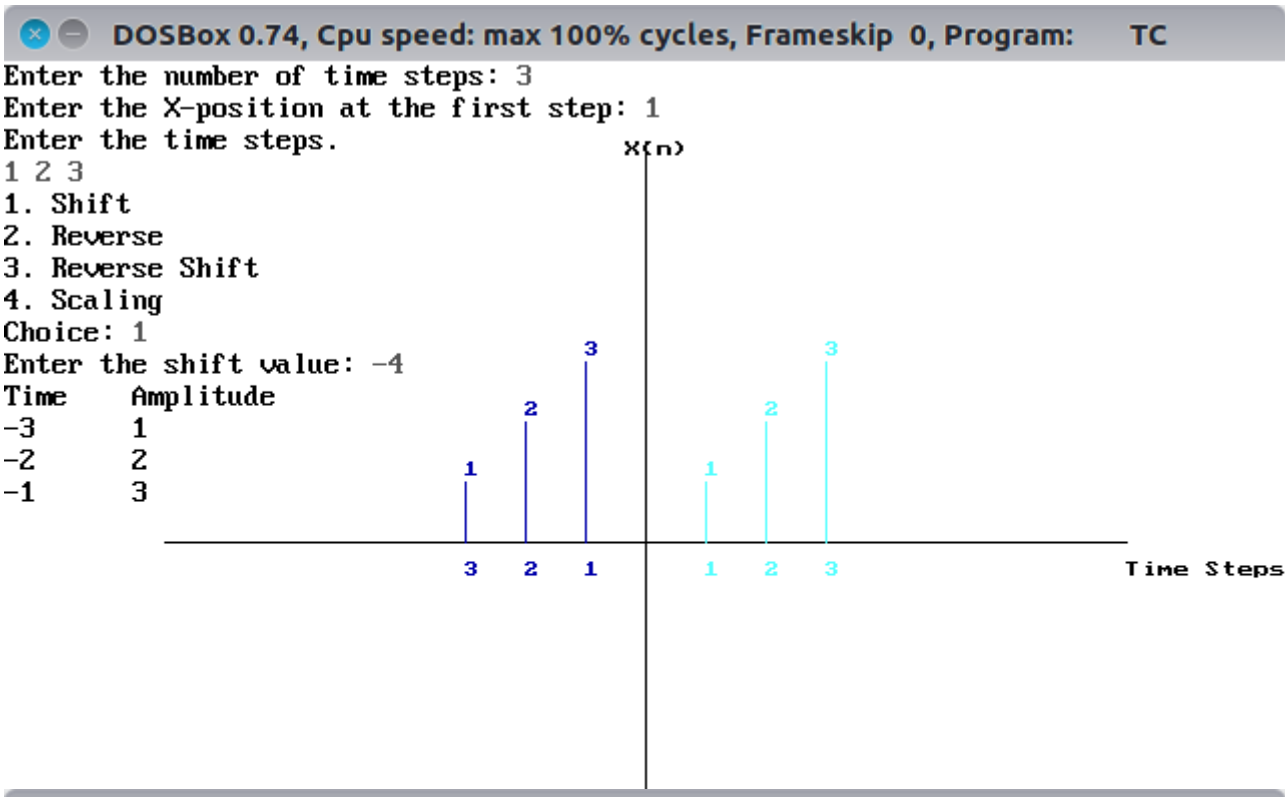
```
	int new_y = origY - 30*val;
	char x_label[2], y_label[2];
	y_label[0] = (char) (abs(val) + 48);
	y_label[1] = '\0';
	x_label[0] = (char)(abs(x)+48);
	x_label[1] = '\0';
	line(new_x, origY, new_x, new_y);
	outtextxy(new_x, new_y - 10, y_label);
	outtextxy(new_x, origY + 10, x_label);
}

void show_graph(int x[2][50], int l, int colour) {
	int i;
	if (colour == 0)
		setcolor(RED);
	else
		setcolor(YELLOW);
	for (i = 0; i < l; i++)
		impulse(x[0][i], x[1][i]);
	setcolor(WHITE);
}

void shift(int x[2][50], int l, int k, int reverse, int result[2][50]) {
	int i;
	for (i = 0; i < l; i++) {
		result[1][i] = x[1][i];
		if (!reverse)
			result[0][i] = x[0][i] + k;
		else
			result[0][i] = x[0][i] - k;
	}
}

void reverse(int x[2][50], int l, int result[2][50]) {
	int i;
	for (i = 0; i < l; i++) {
		result[0][i] = -x[0][l-i-1];
		result[1][i] = x[1][l-i-1];
	}
}

void scale(int x[2][50], int l, int lambda, int result[2][50]) {
	int i;
	for (i = 0; i < l; i++) {
		result[1][i] = x[1][i] * lambda;
		result[0][i] = x[0][i];
	}
}

void add(int x[2][50], int l, int result[2][50]) {
	int y[2][50], m;
	int i, loc;
```

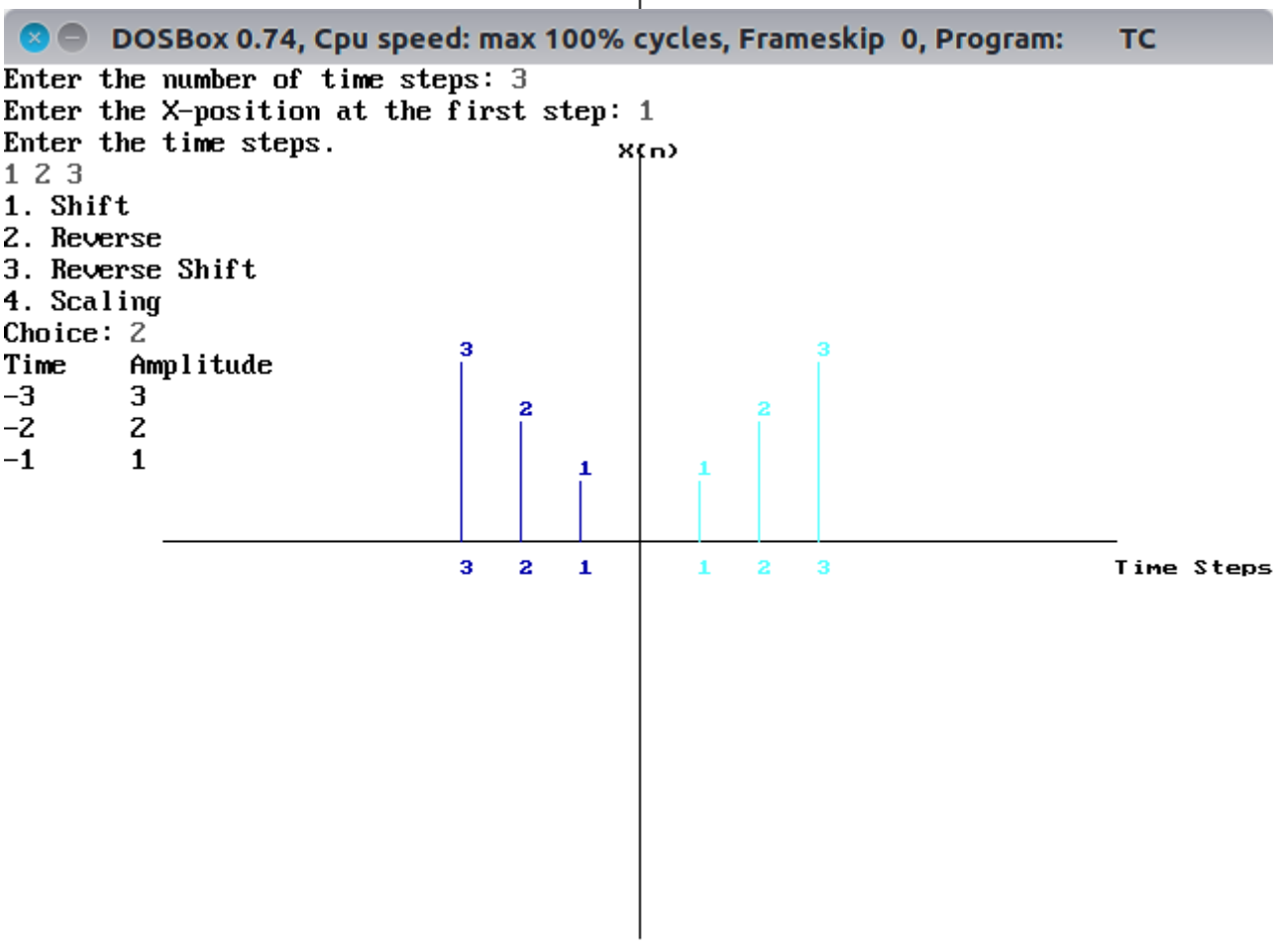
```
	int min, max;
	printf("Enter the number of time steps: ");
	scanf("%d", &m);
	printf("Enter the position of the first time step: ");
	scanf("%d", &loc);
	printf("Enter the time steps: ");
	for (i = 0; i < m; i++) {
		scanf("%d", &y[1][i]);
		y[0][i] = loc + i;
	}
	min = x[0][0] <= y[0][0] ? x[0][0] : y[0][0];
	max = x[0][l-1] >= y[0][m-1] ? x[0][l-1] : y[0][m-1];
	for (i = 0; i <= (max - min); i++) {
		result[0][i] = min + i;
		result[1][i] = 0;
		if (min + i >= x[0][0] && min + i <= x[0][l-1])
			result[1][i] += x[1][min + i - x[0][0]];
		if (min + i >= y[0][0] && min + i <= y[0][m-1])
			result[1][i] += y[1][min + i - y[0][0]];
	}
	show_graph(result, max-min+1, 1);
}

void multiply(int x[2][50], int l, int result[2][50]) {
	int y[2][50], m;
	int i, loc;
	int min, max;
	printf("Enter the number of time steps: ");
	scanf("%d", &m);
	printf("Enter the position of the first time step: ");
	scanf("%d", &loc);
	printf("Enter the time steps: ");
	for (i = 0; i < m; i++) {
		scanf("%d", &y[1][i]);
		y[0][i] = loc + i;
	}
	min = x[0][0] <= y[0][0] ? x[0][0] : y[0][0];
	max = x[0][l-1] >= y[0][m-1] ? x[0][l-1] : y[0][m-1];
	for (i = 0; i <= (max - min); i++) {
		result[0][i] = min + i;
		result[1][i] = 0;
		if (min + i >= x[0][0] && min + i <= x[0][l-1] && min + i >= y[0][0] && min + i
<= y[0][m-1])
			result[1][i] += x[1][min + i - x[0][0]] * y[1][min+i-y[0][0]];
	}
	show_graph(result, max-min+1, 1);
}

int main() {
	int gd = DETECT, gm;
	int opt;
	int x[2][50], l, k, rev, result[2][50];
```

```
	int origin, ch;
	int i;
	initgraph(&gd, &gm, "");
	// Drawing X-axis
	axis(0, origX, origY, 240);
	// Drawing Y-axis
	axis(1, origX, origY, 200);
	printf("Enter the number of time steps: ");
	scanf("%d", &l);
	printf("Enter the X-position at the first step: ");
	scanf("%d", &origin);
	printf("Enter the time steps.\n");
	for (i = 0; i < l; i++) {
		x[0][i] = origin + i;
		scanf("%d", &x[1][i]);
	}
	show_graph(x, l, 0);
	printf("1. Shift\n2. Reverse\n3. Reverse Shift\n4. Scaling\n5. Addition\n6.
Multiplication\nChoice: ");
	scanf("%d", &ch);
	switch(ch) {
		case 1:
			printf("Enter the shift value: ");
			scanf("%d", &k);
			shift(x, l, k, 0, result);
			show_graph(result, l, 1);
			break;
		case 2:
			reverse(x, l, result);
			show_graph(result, l, 1);
			break;
		case 3:
			reverse(x, l, result);
			printf("Enter the shift value: ");
			scanf("%d", &k);
			shift(result, l, k, 1, result);
			show_graph(result, l, 1);
			break;
		case 4:
			printf("Enter the scaling value: ");
			scanf("%d", &k);
			scale(x, l, k, result);
			show_graph(result, l, 1);
			break;
		case 5:
			add(x, l, result);
			break;
		case 6:
			multiply(x, l, result);
			break;
		default:
			printf("Enter a valid choice.");
```

```
    }
    getch();
    closegraph();
    return 0;
}
```

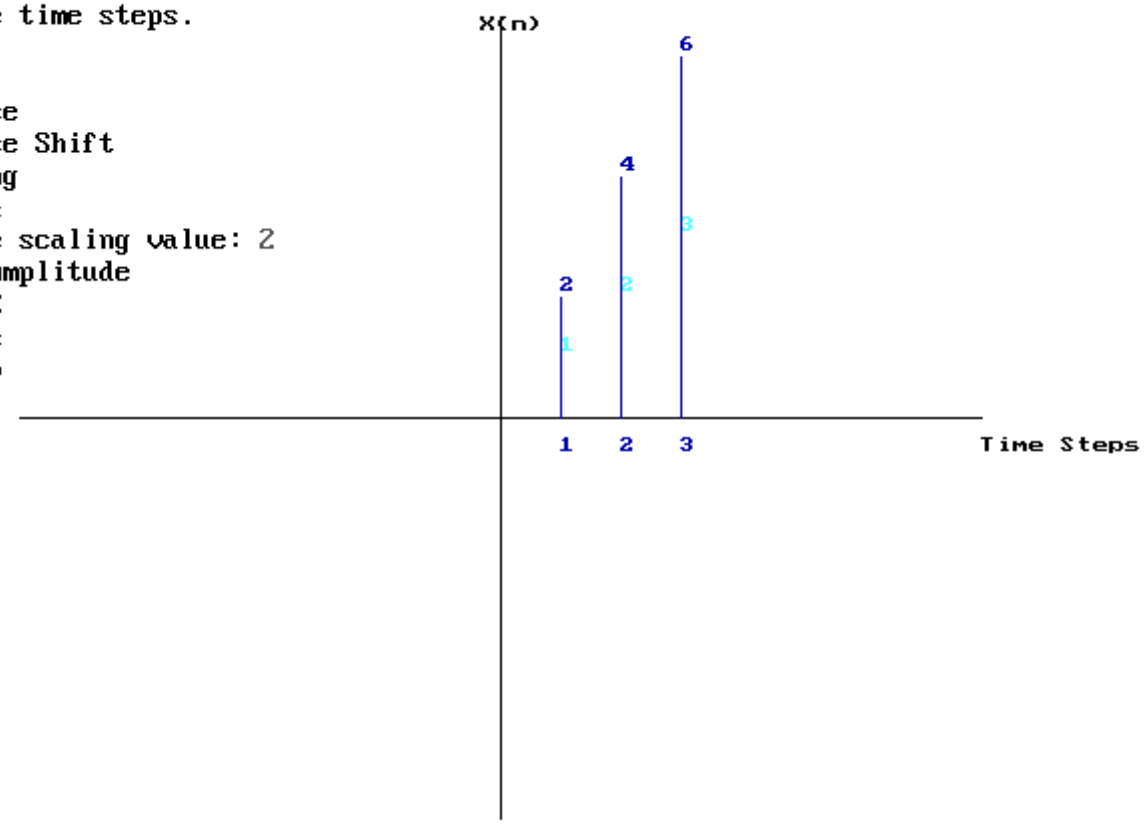
Output:





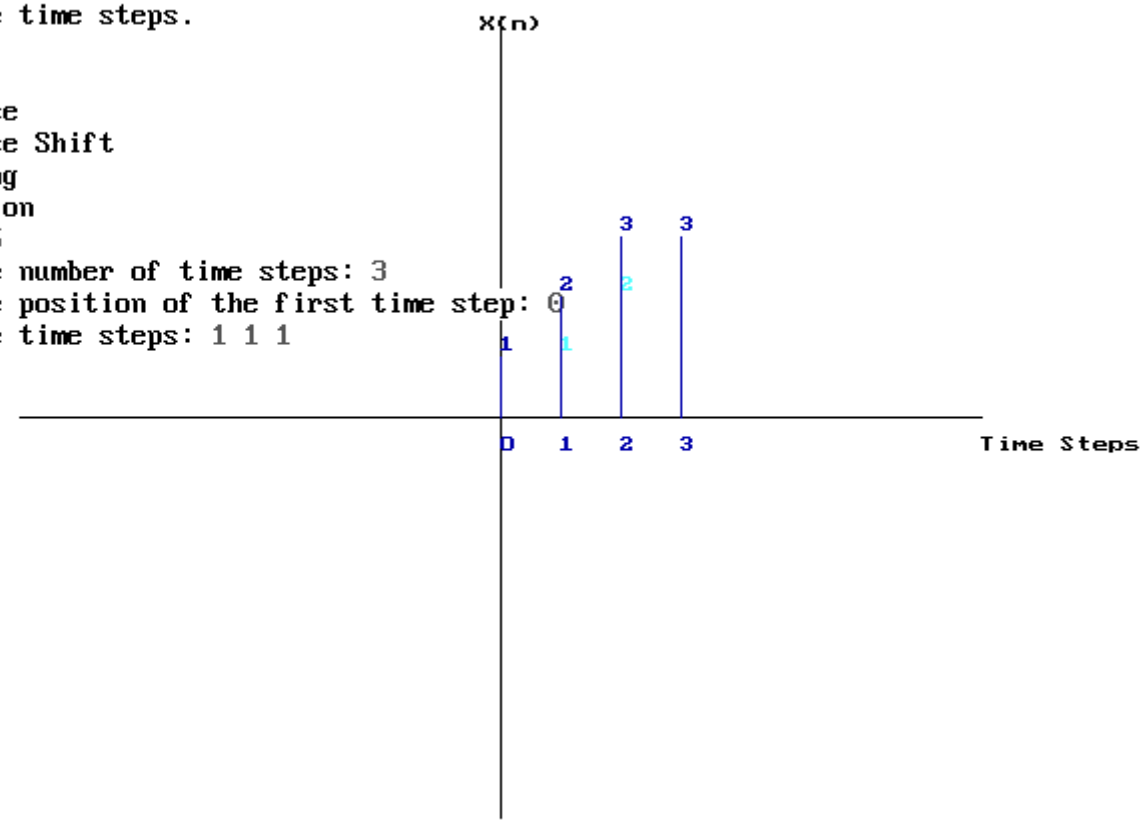
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the number of time steps: 3
Enter the X-position at the first step: 1
Enter the time steps.
1 2 3
1. Shift
2. Reverse
3. Reverse Shift
4. Scaling
Choice: 4
Enter the scaling value: 2
Time    Amplitude
1       2
2       4
3       6
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the number of time steps: 3
Enter the X-position at the first step: 1
Enter the time steps.
1 2 3
1. Shift
2. Reverse
3. Reverse Shift
4. Scaling
5. Addition
Choice: 5
Enter the number of time steps: 3
Enter the position of the first time step: 0
Enter the time steps: 1 1 1
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the number of time steps: 3
Enter the X-position at the first step: 1
Enter the time steps.
1 2 3
1. Shift
2. Reverse
3. Reverse Shift
4. Scaling
5. Addition
6. Multiplication
Choice: 6
Enter the number of time steps: 3
Enter the position of the first time step: 1
Enter the time steps: 1 2 3
```