QUESTION 1

```
#include <stdio.h>
int main()
{
	int batsmen, num_innings;
	printf("Enter number of batsmen: ");
	scanf(" %d", &batsmen);
	printf("Enter number of innings: ");
	scanf(" %d", &num_innings);

	int perf[batsmen][num_innings];
	int total[batsmen];
	int avg[batsmen];
	int high[batsmen];
	int century[batsmen] = {0};
	int half_cen[batsmen] = {0};

	for(int i=0; i<batsmen;i++)
	{
		int temp=0, highest=0;
		for(int j=0; j<num_innings;j++)
		{
			printf("Batsman: %d, Inning: %d, Runs: ", i+1, j+1);
			scanf(" %d", &perf[i][j]);
			temp = temp + perf[i][j];
			if (perf[i][j] > highest)
			{
				high[i] = perf[i][j];
			}
			if (perf[i][j] >= 100)
			{
				century[i] = century[i] + 1;
			}
			else if(perf[i][j] >= 50)
			{
				half_cen[i] = half_cen[i] + 1;
			}
		}
		total[i] = temp;

		avg[i] = temp/num_innings;
	}

	for (int i=0; i<batsmen ; i++)
```

```
        {
                printf("Batsmen %d\n", i+1);
                printf("Total runs scored: %d\n", total[i]);
                printf("Avergae runs scored per inning: %d\n", avg[i]);
                printf("Highest runs scored in a inning: %d\n", high[i]);
                printf("Number of Centuries: %d\n", century[i]);
                printf("Number of Half Centuries: %d\n\n", half_cen[i]);
        }
}
```

```
C:\Users\mabhu\Downl

Enter number of batsmen: 5
Enter number of innings: 5
Batsman: 1, Inning: 1, Runs: 145
Batsman: 1, Inning: 2, Runs: 90
Batsman: 1, Inning: 3, Runs: 45
Batsman: 1, Inning: 4, Runs: 24
Batsman: 1, Inning: 5, Runs: 32
Batsman: 2, Inning: 1, Runs: 27
Batsman: 2, Inning: 2, Runs: 5
Batsman: 2, Inning: 3, Runs: 68
Batsman: 2, Inning: 4, Runs: 66
Batsman: 2, Inning: 5, Runs: 93
Batsman: 3, Inning: 1, Runs: 76
Batsman: 3, Inning: 2, Runs: 115
Batsman: 3, Inning: 3, Runs: 70
Batsman: 3, Inning: 4, Runs: 30
Batsman: 3, Inning: 5, Runs: 146
Batsman: 4, Inning: 1, Runs: 46
Batsman: 4, Inning: 2, Runs: 53
Batsman: 4, Inning: 3, Runs: 16
Batsman: 4, Inning: 4, Runs: 65
Batsman: 4, Inning: 5, Runs: 36
Batsman: 5, Inning: 1, Runs: 90
Batsman: 5, Inning: 2, Runs: 122
Batsman: 5, Inning: 3, Runs: 126
Batsman: 5, Inning: 4, Runs: 78
Batsman: 5, Inning: 5, Runs: 116
Batsmen 1
Total runs scored: 336
Avergae runs scored per inning: 67
Highest runs scored in a inning: 32
Number of Centuries: 1
Number of Half Centuries: 1

Batsmen 2
Total runs scored: 259
Avergae runs scored per inning: 51
Highest runs scored in a inning: 93
Number of Centuries: 0
Number of Half Centuries: 3

Batsmen 3
Total runs scored: 437
Avergae runs scored per inning: 87
Highest runs scored in a inning: 146
Number of Centuries: 2
Number of Half Centuries: 2
```

```
C:\Users\mabhu\Downl

Batsmen 4
Total runs scored: 216
Avergae runs scored per inning: 43
Highest runs scored in a inning: 36
Number of Centuries: 0
Number of Half Centuries: 2

Batsmen 5
Total runs scored: 532
Avergae runs scored per inning: 106
Highest runs scored in a inning: 116
Number of Centuries: 3
Number of Half Centuries: 2


--------------------------------
Process exited after 125.8 seconds with return v
alue 0
Press any key to continue . . .
```

QUESTION 2

```
#include <stdio.h>
int main()
{
	int row, col, len;
	int large[3]= {0}; //[index row, index col, len]
	printf("Enter rows: ");
	scanf(" %d", &row);
	printf("Enter columns: ");
	scanf(" %d", &col);

	int matrix[row][col];

	printf("Enter Matrix\n");
	for (int i=0;i<row;i++)
	{
		for (int j=0;j<col;j++)
		{
			printf("Row: %d, Column: %d, Value: ", i+1, j+1);
			scanf("%d", &matrix[i][j]);
			while (matrix[i][j]!=0 && matrix[i][j]!=1)
```

```
                {
                        printf("Try Again: ");
                        scanf("%d", &matrix[i][j]);
                }
        }
}

for (int i=0;i<row;i++)
{
        for (int j=0;j<col;j++)
        {
                int a=i, b=j, row_len=0, col_len=0,sq_len=0;
                if (matrix[a][b]==1)
                {
                        while (matrix[i][b+1]==1 && matrix[a+1][j]==1)
                        {
                                len = a+2-i;
                                a = a+1;
                                b = b+1;
                        }
                }

                for (int k=i; k<i+len;k++)
                {
                        for (int l=j;l<j+len;l++)
                        {
                                if (matrix[k][l]==1)
                                {
                                        row_len = k-i+1;
                                        col_len = l-j+1;
                                }
                        }
                }

                if (row_len>1 && col_len>1)
                {
                        if (row_len == col_len)
                        {
                                sq_len = row_len;
                        }
                        else if(row_len > col_len)
                        {
                                sq_len = col_len;
                        }
```

```
                    else if(row_len < col_len)
                    {
                        sq_len = row_len;
                    }

                    if (sq_len > large[2])
                    {
                        large[0] = i;
                        large[1] = j;
                        large[2] = sq_len;
                    }
                    printf("Square Submatrix starting at index (%d, %d) with length
%d\n",i,j, sq_len);
                }
            }
        }
        printf("\nLargest Square Submatrix starting at index (%d, %d) with length
%d\n",large[0],large[1],large[2]);
}
```

```
C:\Users\mabhu\Downloads\a

Enter rows: 5
Enter columns: 5
Enter Matrix
Row: 1, Column: 1, Value: 0
Row: 1, Column: 2, Value: 1
Row: 1, Column: 3, Value: 1
Row: 1, Column: 4, Value: 1
Row: 1, Column: 5, Value: 0
Row: 2, Column: 1, Value: 1
Row: 2, Column: 2, Value: 1
Row: 2, Column: 3, Value: 1
Row: 2, Column: 4, Value: 1
Row: 2, Column: 5, Value: 1
Row: 3, Column: 1, Value: 1
Row: 3, Column: 2, Value: 1
Row: 3, Column: 3, Value: 1
Row: 3, Column: 4, Value: 1
Row: 3, Column: 5, Value: 1
Row: 4, Column: 1, Value: 0
Row: 4, Column: 2, Value: 1
Row: 4, Column: 3, Value: 1
Row: 4, Column: 4, Value: 1
Row: 4, Column: 5, Value: 1
Row: 5, Column: 1, Value: 0
Row: 5, Column: 2, Value: 1
Row: 5, Column: 3, Value: 1
Row: 5, Column: 4, Value: 1
Row: 5, Column: 5, Value: 1
Square Submatrix starting at index (0, 1) with length 3
Square Submatrix starting at index (0, 2) with length 2
Square Submatrix starting at index (0, 3) with length 2
Square Submatrix starting at index (0, 4) with length 2
Square Submatrix starting at index (1, 0) with length 2
Square Submatrix starting at index (1, 1) with length 4
Square Submatrix starting at index (1, 2) with length 3
Square Submatrix starting at index (1, 3) with length 2
Square Submatrix starting at index (2, 0) with length 3
Square Submatrix starting at index (2, 1) with length 3
Square Submatrix starting at index (2, 2) with length 3
Square Submatrix starting at index (2, 3) with length 2
Square Submatrix starting at index (3, 0) with length 2
Square Submatrix starting at index (3, 1) with length 2
Square Submatrix starting at index (3, 2) with length 2
Square Submatrix starting at index (3, 3) with length 2

Largest Square Submatrix starting at index (1, 1) with length 4

--------------------------------
Process exited after 116.1 seconds with return value 0
Press any key to continue . . .
```

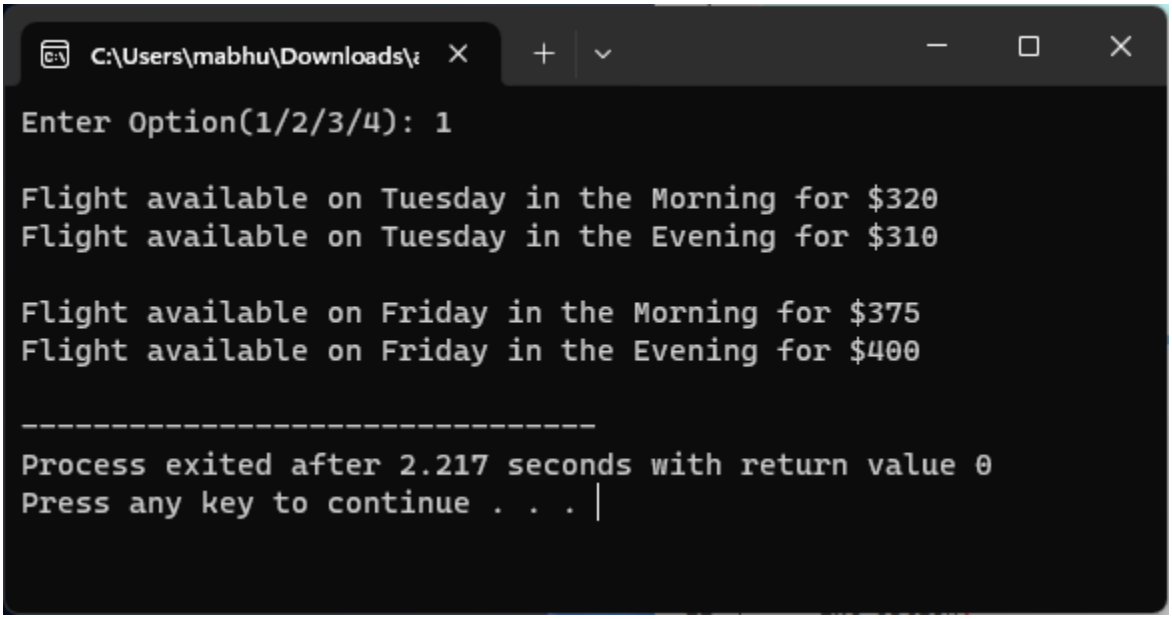
QUESTION 3

```
#include <stdio.h>

int main()
{
	int flight[5][4] = {
	{1,300,0,-1},
	{1,320,1,310},
	{0,-1,1,280},
	{1,380,0,-1},
	{1,375,1,400}
	};

	char days[5][10] = {"Monday","Tuesday","Wednesday","Thursday","Friday"};

	char timeofday[2][8] = {"Morning", "Evening"};
	int i,j,price,k;

	int option;
	printf("Enter Option(1/2/3/4): ");
	scanf(" %d", &option);

	switch(option)
	{
		case 1:
			for(i=0; i<5 ; i++)
			{
				j=0;
				if (flight[i][j]==1 && flight[i][j+2]==1)
				{
					printf("\nFlight available on %s in the %s for $%d\n",days[i]
,timeofday[0] ,flight[i][j+1]);
					printf("Flight available on %s in the %s for $%d\n",days[i]
,timeofday[1] ,flight[i][j+2+1]);
				}
			}

			break;

		case 2:
			j=0;
			price=-1;
```

```
                printf("\n");
                for (int i=0;i<5;i++)
                {
                        if (flight[i][j]==1)
                        {
                                if(price==-1)
                                {
                                        price = i;
                                }
                                printf("Flight available on %s in the %s for $%d\n",days[i]
,timeofday[j] ,flight[i][j+1]);
                                if (flight[i][j+1] < flight[price][j+1])
                                {
                                        price = i;
                                }
                        }
                }
                printf("\nBest Option: Flight available on %s in the %s for $%d\n",days[price]
,timeofday[j] ,flight[price][j+1]);

                break;

        case 3:
                j=2;
                price=-1;

                printf("\n");
                for (i=0;i<5;i++)
                {
                        if (flight[i][j]==1)
                        {
                                if (price==-1)
                                {
                                        price = i;
                                }
                                printf("Flight available on %s in the %s for $%d\n",days[i]
,timeofday[j-1] ,flight[i][j+1]);
                                if (flight[i][j+1] < flight[price][j+1])
                                {
                                        price = i;
                                }
                        }
                }
```

```
                        printf("\nBest Option: Flight available on %s in the %s for $%d\n",days[price]
,timeofday[j-1] ,flight[price][j+1]);

                        break;

                case 4:
                        int day;
                        printf("\n1.Monday\n2.Tuesday\n3.Wednesday\n4.Thursday\n5.Friday\n");
                        printf("Enter your prefered Day: ");
                        scanf(" %d", &day);
                        day = day-1;

                        printf("\n");
                        for (j=0, k=0; j<=2;j+=2,k++)
                        {
                                if (flight[day][j]==1)
                                {
                                        printf("Flight available on %s in the %s for $%d\n",days[day]
,timeofday[k] ,flight[day][j+1]);
                                }
                        }

                        break;
        }

}
```

PART 1



PART 2

```
C:\Users\mabhu\Downloads\a

Enter Option(1/2/3/4): 2

Flight available on Monday in the Morning for $300
Flight available on Tuesday in the Morning for $320
Flight available on Thursday in the Morning for $380
Flight available on Friday in the Morning for $375

Best Option: Flight available on Monday in the Morning for $300

--------------------------------
Process exited after 1.083 seconds with return value 0
Press any key to continue . . .
```

PART 3

```
C:\Users\mabhu\Downloads\a

Enter Option(1/2/3/4): 3

Flight available on Tuesday in the Evening for $310
Flight available on Wednesday in the Evening for $280
Flight available on Friday in the Evening for $400

Best Option: Flight available on Wednesday in the Evening for $280

--------------------------------
Process exited after 1.181 seconds with return value 0
Press any key to continue . . .
```

PART 4

```
C:\Users\mabhu\Downloads\a

Enter Option(1/2/3/4): 4

1.Monday
2.Tuesday
3.Wednesday
4.Thursday
5.Friday
Enter your prefered Day: 2

Flight available on Tuesday in the Morning for $320
Flight available on Tuesday in the Evening for $310

--------------------------------
Process exited after 3.781 seconds with return value 0
Press any key to continue . . .
```

QUESTION 4

```
#include <stdio.h>

int main()
{
	char maze[5][5];

	printf("Input original maze:\n");
	for(int x=0; x<5;x++)
	{
		printf("Row %d\n", x+1);
		for (int y=0;y<5;y++)
		{
			scanf(" %c", &maze[x][y]);
		}
	}

	int found =0;
	int a,b;
	int i=0, j=0;
	printf("\nOutput\n");
	printf("%d,%d\n",i,j);
	while (found!=1)
```

```
        {
                if (maze[i][j+1] == 'O' && maze[i+1][j] == 'O')
                {
                        if (maze[i][j+2] == 'W' || maze[i+2][j] == 'W')
                        {
                                if (maze[i][j+2] == 'W')
                                {
                                        i = i+1;
                                }
                                else if(maze[i+2][j] == 'W')
                                {
                                        j = j+1;
                                }
                                printf("%d,%d\n",i,j);
                        }
                }

                if (maze[i][j+1] == 'O' || maze[i+1][j] == 'O')
                {
                        if (maze[i][j+1] == 'O')
                        {
                                j = j+1;
                        }
                        else if (maze[i+1][j] == 'O')
                        {
                                i = i+1;
                        }
                        printf("%d,%d\n",i,j);
                }
                else if (maze[i+1][j] == 'E' || maze[i][j+1] == 'E')
                {
                        found = 1;
                        if (maze[i+1][j] == 'E')
                        {
                                i = i+1;
                        }
                        else if(maze[i][j+1] == 'E')
                        {
                                j = j+1;
                        }
                        printf("%d,%d\n",i,j);
                }
        }
}
```

```
C:\Users\mabhu\Downloads\a

Input original maze:
Row 1
S
O
O
W
W
Row 2
O
W
O
W
W
Row 3
W
O
O
W
O
Row 4
W
W
O
W
O
Row 5
W
W
O
E
W

Output
0,0
0,1
0,2
1,2
2,2
3,2
4,2
4,3

--------------------------------
Process exited after 41.57 seconds with return value 0
Press any key to continue . . .
```

QUESTION 5

```
#include <stdio.h>

int main()
{
	int n;
	printf("Enter n: ");
	scanf(" %d", &n);
	int n_cube = n*n*n;
	int a_cube,b_cube,c_cube,d_cube;

	printf("\nAll Ramanujan-Hardy numbers less than %d^3\n\n", n);

	for (int a=1; a*a*a < n_cube; a++)
	{
		a_cube = a*a*a;
		for(int b=a; a_cube + b*b*b < n_cube ; b++)
		{
			b_cube = b*b*b;
			for(int c=a+1; c*c*c < n_cube ; c++)
			{
				c_cube = c*c*c;
				for(int d=c; c_cube + d*d*d < n_cube; d++)
				{
					d_cube = d*d*d;
					if (a!=c && b!=d)
					{
						if ((a_cube + b_cube) == (c_cube + d_cube))
						{
							printf("%d = %d^3 +%d^3 = %d^3 + %d^3\n",
a_cube+b_cube,a,b,c,d);
						}
					}
				}
			}
		}
	}
}
```

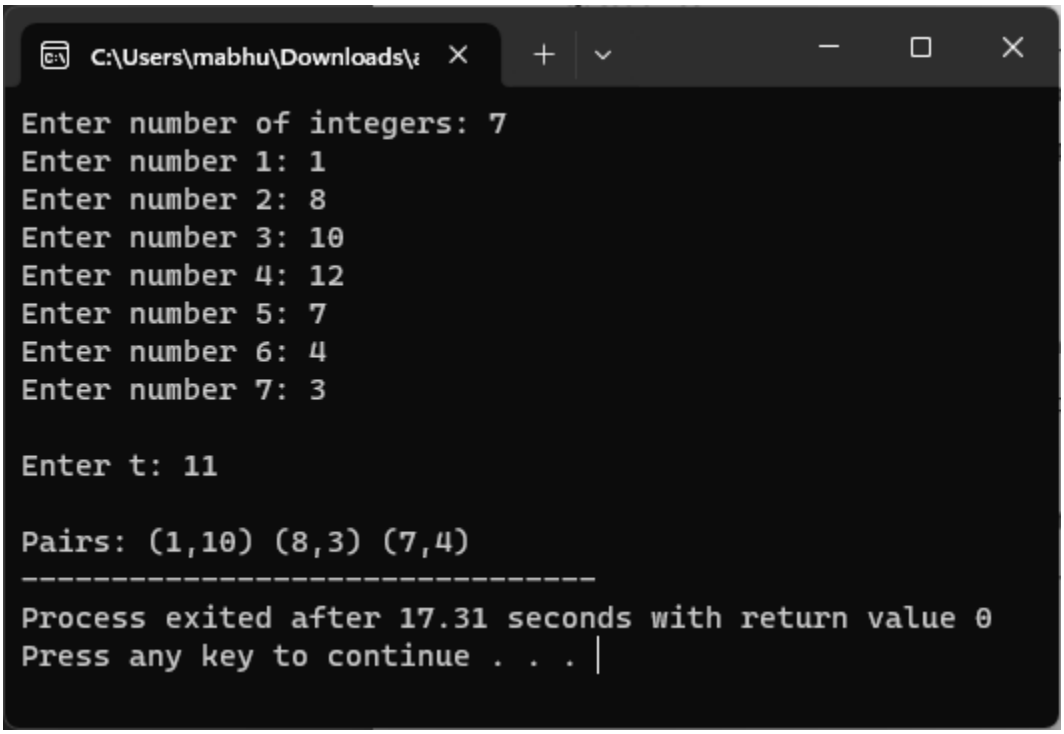
```
C:\Users\mabhu\Downloads\a

Enter n: 50

All Ramanujan-Hardy numbers less than 50^3

1729 = 1^3 +12^3 = 9^3 + 10^3
4104 = 2^3 +16^3 = 9^3 + 15^3
13832 = 2^3 +24^3 = 18^3 + 20^3
39312 = 2^3 +34^3 = 15^3 + 33^3
46683 = 3^3 +36^3 = 27^3 + 30^3
32832 = 4^3 +32^3 = 18^3 + 30^3
110656 = 4^3 +48^3 = 36^3 + 40^3
110808 = 6^3 +48^3 = 27^3 + 45^3
40033 = 9^3 +34^3 = 16^3 + 33^3
20683 = 10^3 +27^3 = 19^3 + 24^3
65728 = 12^3 +40^3 = 31^3 + 33^3
64232 = 17^3 +39^3 = 26^3 + 36^3

--------------------------------
Process exited after 2.202 seconds with return value 0
Press any key to continue . . .
```

n	n_cube	a	a_cube	b	b_cube	c	c_cube	d	d_cube	OUTPUT
25	15625									
		1	1	12	1728	9	729	10	1000	1729 = 1^3 +12^3 = 9^3 + 10^3
		2	1	16	4096	9	729	15	3375	4104 = 2^3 +16^3 = 9^3 + 15^3
		2	8	24	13824	18	5832	20	8000	13832 = 2^3 +24^3 = 18^3 + 20^3

QUESTION 6

```
#include <stdio.h>
int main()
{
	int len, t, temp;

	printf("Enter number of integers: ");
	scanf(" %d", &len);

	int num[len];
	for (int i=0; i<len; i++)
	{
		printf("Enter number %d: ", i+1);
		scanf(" %d", &num[i]);
```

```
        }

        printf("\nEnter t: ");
        scanf(" %d", &t);

        printf("\nPairs: ");
        for (int i=0; i<len;i++)
        {
                for (int j=i; j<len ;j++)
                {
                        if (num[i] + num[j] == t)
                        {
                                printf("(%d,%d) ", num[i],num[j]);
                        }
                }
        }

}
```



QUESTION 7

```
#include <stdio.h>

int main()
{
        int shirts[12][2]= { //age, price
```

```
{23,19},
{15,17},
{10,20},
{20,29},
{15,18},
{23,25},
{10,22},
{20,15},
{10,26},
{23,27},
{20,30},
{15,21}
};


printf("Before Sorting\n");
for (int i=0;i<12;i++)
{
	printf("Age: %d, Price:%d\n", shirts[i][0],shirts[i][1]);
}
printf("\n");
int boundary=11;
while (boundary>0)
{
	for(int b=0; b<boundary;b++)
	{
		if (shirts[b][0] > shirts[b+1][0])
		{
			int temp1 = shirts[b][0];
			shirts[b][0] = shirts[b+1][0];
			shirts[b+1][0] = temp1;

			int temp2 = shirts[b][1];
			shirts[b][1] = shirts[b+1][1];
			shirts[b+1][1] = temp2;
		}
	}
	boundary=boundary-1;
}
printf("After Sorting with respect to Age in ascending order\n");
for (int i=0;i<12;i++)
{
	printf("Age: %d, Price:%d\n", shirts[i][0],shirts[i][1]);
}
```

```
	printf("\n");

	boundary=11;

	while (boundary>0)
	{
		for(int x=0;x<12;x++)
		{
			if((shirts[x][0] == shirts[x+1][0]) && (shirts[x][1] < shirts[x+1][1]))
			{
				int temp2 = shirts[x][1];
				shirts[x][1] = shirts[x+1][1];
				shirts[x+1][1] = temp2;
			}
		}
		boundary = boundary-1;
	}

	printf("After Sorting with respect to Age in acsending order and Price within Ages in descending
order\n");
	for (int i=0;i<12;i++)
	{
		printf("Age: %d, Price:%d\n", shirts[i][0],shirts[i][1]);
	}
	printf("\n");
}
```

```
C:\Users\mabhu\Downloads\a

Before Sorting
Age: 23, Price:19
Age: 15, Price:17
Age: 10, Price:20
Age: 20, Price:29
Age: 15, Price:18
Age: 23, Price:25
Age: 10, Price:22
Age: 20, Price:15
Age: 10, Price:26
Age: 23, Price:27
Age: 20, Price:30
Age: 15, Price:21

After Sorting with respect to Age in ascending order
Age: 10, Price:20
Age: 10, Price:22
Age: 10, Price:26
Age: 15, Price:17
Age: 15, Price:18
Age: 15, Price:21
Age: 20, Price:29
Age: 20, Price:15
Age: 20, Price:30
Age: 23, Price:19
Age: 23, Price:25
Age: 23, Price:27

After Sorting with respect to Age in acsending order and Price within Ages in descending order
Age: 10, Price:26
Age: 10, Price:22
Age: 10, Price:20
Age: 15, Price:21
Age: 15, Price:18
Age: 15, Price:17
Age: 20, Price:30
Age: 20, Price:29
Age: 20, Price:15
Age: 23, Price:27
Age: 23, Price:25
Age: 23, Price:19


--------------------------------
Process exited after 0.7613 seconds with return value 0
Press any key to continue . . .
```