Program:

```
computer$ gcc -o instruments instruments.c
computer$ ./instruments instruments.txt
**Least expensive: cymbals at $49.00. It is from the percussion family.
```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct instr{

	char name[25];
	float price;
	char family;

};

typedef struct instr instrument;

void read_file(int nums, instrument* n, char *c) /*passing in a pointer at an array of structs*/
{
	FILE*fp=fopen(c, "r+");

	int i=0;
	char line[100];
	char*token;

	while(i<nums)
	{
		fgets(line, 100, fp);

		token=strtok(line, ",");
		strcpy(n->name, token);

		token=strtok(NULL, ",");
		n->price=atof(token);

		token=strtok(NULL, ",\n");
		n->family=token[0];
		n++;
		i++;



	}

	fclose(fp);
}
```

```
/*send back index of least expensive instrument*/
int least_expensive(instrument *ptr, int nums)
{
	float hold=ptr[0].price, current=0;
	int i;
	int index=0; /*starting*/

	for(i=1;i<nums;i++)
	{
		current=ptr[i].price;
		if(current<hold)
		{
			hold=ptr[i].price;
			index=i;
		}
	}

	return index;
}

void print_info(instrument n) /*just passing one struct*/
{
	char family[15];

	switch(n.family)
	{
		case'k':
			strcpy(family, "keyboards");
			break;
		case's':
			strcpy(family, "strings");
			break;
		case'b':
			strcpy(family, "brass");
			break;
		case'p':
			strcpy(family, "percussion");
			break;
		case'w':
			strcpy(family, "woodwind");
			break;
		default:
			strcpy(family,"unknown");
	}

	printf("\n**Least expensive: %s at $%.2f. It is from the %s family.\n\n",n.name,n.price,
family);
}
```

```
int main (int argc, char**argv) {


    instrument symphony[18];
    instrument *ptr=symphony;

    read_file(18, ptr,argv[1]);

    int n=least_expensive(ptr, 18);
    print_info(symphony[n]);

    return 0;
}
```

Program 1:

```
computer$ gcc price.c
computer$ ./a.out
How many items to buy? 2
Enter price of item 1: $4.55
Enter price of item 2: $5.99
***Price list:***
$4.55
$5.99
```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>


void enter_prices(float*f, int n)
{
  int i;
  for(i=0;i<n;i++)
  {
        printf("Enter price of item %d: $", (i+1));
        scanf("%f", f);
        f++; /*pointer is changing (in function)*/
  }
}



void print_prices(float*f, int n)
{
```

```
  int i;
  printf("***Price list:***\n");
  for(i=0;i<n;i++)
  {
        printf("$%.2f\n", f[i]); /*using the index method-pointer not changing (in function)*/
  }
}



int main(int argc, char **argv)

{

  int n;
  printf("How many items to buy? ");
  scanf("%d", &n);

  float*prices=malloc(sizeof(float)*n); /*dynamically allocating*/

  if(prices==NULL)
  {
        printf("Memory not allocated.  Exiting...\n");
  }

  else
  {
        enter_prices(prices, n);
        print_prices(prices, n);

        free(prices); /*don't forget to free your memory*/
  }

}
```