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

//Membuat struct
struct stack{
    char name[55];
              char type[55];
              int price;
              int quantity;
              struct stack *prev;
}*top=NULL, *curr=NULL;;

//Fungsi yang digunakan
void display();
void show();
void add();
void deleteData();

//Fungsi utama
int main()
{
    display();
    return 0;
}

//Menu – menu program
void display(menu){
    printf("COOL COMPUTER ADMINISTRATOR\n");
    printf("***************************\n\n");
    printf("1. Item List\n");
    printf("2. Add <PUSH> Recently Added Item\n");
```

```
  printf("3. Delete <POP> Recently Added Item\n");
  printf("4. Exit\n\n");
  printf(">> Input your choice : ");
  scanf("%d", &menu);
  if(menu == 1){
    show();
  }
  else if(menu == 2){
    add();
  }
  else if(menu == 3){
    deleteData();
  }
  else if(menu == 4){
    exit(0);
  }
  else{
    printf("Option Not Found\n");
  }
}

//Fungsi menampilkan data
void show(){
  int ctr=0;
  struct  stack*curr = top;
  if(top == NULL){
    printf("\n\n\t\t  --- ITEM LIST ---\n\n");
    printf("-------+-------------------+--------------+----------+-------\n");
    printf("| No.  | Name              | Type         | Quantity | Price |\n");
    printf("-------+-------------------+--------------+----------+-------\n");
    printf("-------+-------------------+--------------+----------+-------\n\n");
```

```
        return display();
    }
    else{
        printf("\n\n\t\t  --- ITEM LIST ---\n\n");
        printf("-------+---------------------------+--------------------+----------------+-----------\n");
        printf("| Desc | Name                      | Type               | Quantity       |   Price  |\n");
        printf("-------+---------------------------+--------------------+----------------+-----------\n");


        while(curr!=NULL){
            ctr++;
            printf("| %d.   | %-20s      | %-12s       |%3d             |$%4d     | \n",ctr,curr->name,curr-
>type,curr->quantity,curr->price);
            curr = curr->prev;
        }
        printf("-------+---------------------------+--------------------+----------------+-----------\n");
        return display();
    }
}


//Fungsi menambah data
void add(){
    char name[55], type[55];
    int price,quantity;
    printf("Input Name of The New Item [3..20]: ");
    getchar();
    scanf("%[^\n]",name);
    if(strlen(name) < 3 || strlen(name) > 20){
        printf("Name Not Found \n");
    }
    else{
        printf("Input Type  of The New Item [processor/graphic card/memory]: ");
        getchar();
        scanf("%[^\n]",type);
        if(strcmp("processor",type)==0 ||strcmp("graphic card",type)==0 ||
strcmp("memory",type)==0){
            printf("Input Quantity of The New Item [1...20]: ");
            scanf("%d",&quantity);
            if(quantity < 1 || quantity > 20){
                printf("Quantity Not Found\n");
            }
        else{
            printf("Input Price of The New Item [$1...$1000]: $");
            scanf("%d",&price);
            if(price<1 || price > 1000){
                printf("Price Not Found\n");
            }
            else{
                PUSH(name,type,quantity,price);
                printf("\n\n --- Add New Item Success ---\n\n");
            }
        }
        }
        else{
            printf("Type Not Found\n\n");
        }
    }
    return display();
}

//Fungsi menghapus data
void deleteData(){
    POP();
    puts("");
    return display();
}

//Untuk menjalankan fungsi add
void PUSH(const char *name, const char *type,int quantity,int price){
    curr=(struct stack*)malloc(sizeof(struct stack));
    curr->quantity=quantity;
    curr->price=price;
    strcpy(curr->name,name);
    strcpy(curr->type,type);
    curr->prev=NULL;
    if(top==NULL){
        top=curr;
    }
    else{
        curr->prev=top;
        top=curr;
    }
}

//Untuk menjalankan fungsi delete
void POP(){
    struct stack *curr=top;
    if(top==NULL){
        printf("No Item Available\n\n");
    }
    else if(top!=NULL){
        printf("The recently added item < %s - %s > is being delete.\n",curr->name,curr->type);
        curr=top;
        top=top->prev;
        free(curr);
    }
}

void X(){
    curr=top;
    while(curr!=NULL){
        top=top->prev;
        free(curr);
        curr=top;
    }
}
```