Typical answer for homework #2

Part1: Write a program that performs the previous tasks using a **single dynamic array** of size N.

Answer 1: **(10 Pts)**

```
#include<stdio.h>
#include<stdlib.h>
int* intit (int *t,int n)          1,5pts
{
int i;
t=(int*)malloc(n*sizeof(int)) ;
for(i=0;i<n ;i++)
t[i]=i;
return t;}
//----------------
int * settozero(int *t,int n)    1,5pts
{int i ,p,j;
for(i=2;i<n ;i++)
if(t[i])
  for(p=2,j=2*i;j<n;p++,j=p*i)
t[j]=0;
  return t;}
  //-----------------
 void delMulti(int *t,int *n){
4pts
 int i, j=0;
for(i=0;i<*n-1;i++)
{if(t[i]>2)
{t[j]=t[i];
j++;
}
}
*n=j;
t=(int *)realloc(t, *n*sizeof(int));
}
//------------------------
void displayVect(int *t,int n ){   1pts
int i;
for(i=0;i<n;i++)
printf("%d ",t[i]);}
```

```
//************************
int main(){                          2pts
int *t,i,n,j,p;
printf("enter n :");
scanf("%d",&n);
//----------------
t=intit (t,n);
displayVect(t,n);
//---------
t=settozero(t,n);
printf ("\n \n \n result2:\n");
displayVect(t,n);
//----------------
delMulti(t,&n);

printf ("\n \n \n result3 :\n");
displayVect(t,n);
printf ("\n size of vector :%d",n*sizeof(t));
}
```

Part2: rewrite the program using a **linked list**

Answer2: **(10 Pts)**

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node{
int data;
struct node* next;}Node;
typedef Node * List;
//_____________________________
void display (List l )
{    if (l){
    printf("%d ",l->data);
    display(l->next);}
    else printf(".");
}
//_____________________________________
int append (List *aHead, int d )
{   List e,t;
    e=(List)malloc(sizeof(Node));
    if(!e) return 0;
    else
    {   e->data=d;
    e->next=NULL;
    if (*aHead==NULL)
    *aHead=e;
    else{
    t=*aHead;
    while(t->next)
    t=t->next;
    t->next=e;}
    return 1;}
}
//_____________________________________
int delhead( List *l)
{     List p;
      if (!*l) return 0;
      else
      {p=*l;
      *l=p->next;
      free(p);
      return 1;}
}
//_____________________________________
```

```
int del(List *l, int d)                    (5pts)
{
    List t,p=*l;
    while((*l)->data==d)
    delhead(l);

    while(p->next)
    { if (p->next->data==d)
    {t=p->next ;
    p->next=p->next->next;
    free(t);
    }
    else p=p->next ;}
    return t;
}
 //_____________________________
int main()
{   List t=NULL, q=NULL,v;
int i,n;
printf("Enter n : ");
scanf("%d",&n);;

for(i=1;i<=n;i++)
append(&t,i);                              01pts

printf("First step :\n");
display(t);
for(q=t->next;q->next!=0;q=q->next)
for(v=q->next;v!=0;v=v->next)
if(v->data)
if ((v->data)%(q->data)==0)
v->data=0;                                 4ts

printf("\n Second step :\n");
display(t);
for(v=t;v!=0;v=v->next)
del(&t,0);
printf("\n Third step :\n");
display(t);
return 0;
}
```