1.

2	10	6
0		
3		
7		

2.

```
int KarakterTekrariKontrl(char *p,char c)
{
    int i,syc=0;;
    for(i=0; *(p+i)!='\0'; i++)
    {
        if(*(p+i)==c)
            syc++;
    }
    if(syc==0)//Aranan kelime 0 kere tekrar ediyorsa
        return 0;
    else
        return 1;
}
int main()
{
    int bboyut=0,kboyut=0,i,sonuc;
    char *pkelime, *pk, *pb;

    pkelime=malloc(50*sizeof(char));

    printf("Karakter katarini giriniz:");
    //scanf("%s", pkelime);
    gets(pkelime);
    for(i=0; *(pkelime+i)!='\0'; i++)
    {
        //Kucuk harf ise
        if(*(pkelime+i)<='z' && *(pkelime+i)>='a')
        {
            if(kboyut>0)
            {
                sonuc=KarakterTekrariKontrl(pk,*(pkelime+i));
                if(sonuc==0)
                {
                    kboyut++;
                    pk=realloc(pk,kboyut*sizeof(char));
                    *(pk+kboyut-1)=*(pkelime+i);
                }
            }
            else
            {
                kboyut++;
                pk=malloc(kboyut*sizeof(char));
                *(pk+kboyut-1)=*(pkelime+i);
            }

        }//Buyuk harf ise
        else if(*(pkelime+i)<='Z' && *(pkelime+i)>='A')
        {
            if(bboyut>0)
            {
                sonuc=KarakterTekrariKontrl(pb,*(pkelime+i));
                if(sonuc==0)
                {
                    bboyut++;
                    pb=realloc(pb,bboyut*sizeof(char));
                    *(pb+bboyut-1)=*(pkelime+i);
                }
            }
            else
            {
                bboyut++;
                pb=malloc(bboyut*sizeof(char));
                *(pb+bboyut-1)=*(pkelime+i);
            }
        }
    }
    *(pk+kboyut)='\0';
    *(pb+bboyut)='\0';

    printf("\nKucuk harfler: %s\n",pk);
    printf("Buyuk harfler: %s\n",pb);

    return 0;
}
```

3.

```
typedef struct Kurs
{
    int kTip;
    int aidat;
    int sure;
}KT;
typedef struct Abone
{
    int aNo;
    char ad[20];
    int toplamAidat;
    int kSayisi;
    KT *kurslar;

}AB,A;

AB *bilgi_al(int aSayisi);
void borc_hesapla(AB *aboneler, int aSayisi);
void bilgi_goster(AB *aboneler, int aSayisi);

int main()
{
    int aboneSayisi;
    AB *aboneler;
    printf("Kac adet abone girilecek: ");
    scanf("%d", &aboneSayisi);

    //b
    aboneler = bilgi_al(aboneSayisi);
    //c
    aidat_hesapla(aboneler, aboneSayisi);
    //d
   // bilgi_goster(aboneler, aboneSayisi);

    return 0;
}

AB *bilgi_al(int aSayisi)
{
    int i,j,kSayisi;
    AB *aboneler = (AB*)malloc(sizeof(AB)*aSayisi);



    for(i=0;i<aSayisi;i++)
    {
        printf("%d.Abone no: ",i+1);
        scanf("%d", &(aboneler+i)->aNo);

        printf("%d.Abone adi: ",i+1);
        scanf("%s", (aboneler+i)->ad);

        printf("%d.Abone kurs sayisi: ",i+1);
        scanf("%d", &kSayisi);

        (aboneler+i)->kSayisi = kSayisi;  //abonelar[i].fSayisi=fSayisi;
        (aboneler+i)->kurslar = (KT*)malloc(sizeof(KT)*kSayisi);

        for(j=0;j<kSayisi;j++)
        {
            printf("%d.Abone icin %d. kurs tipi: ",i+1, j+1);
            scanf("%d", &((aboneler+i)->kurslar+j)->kTip);

            //scanf("%d", &abonelar[i].faturalar[j].fTip);
            printf("%d.Abone icin %d. kurs aidati: ",i+1, j+1);
            scanf("%d", &((aboneler+i)->kurslar+j)->aidat);

            printf("%d.Abone icin %d. kurs suresi: ",i+1, j+1);
            scanf("%d", &((aboneler+i)->kurslar+j)->sure);
        }
    }

    return aboneler;
}

void aidat_hesapla(AB *aboneler, int aSayisi)
{
    int i,j,kSayisi,tBorc;

    for(i=0;i<aSayisi;i++)
    {
        kSayisi = (aboneler+i)->kSayisi;
        tBorc = 0;
        for(j=0;j<kSayisi;j++)
        {
            tBorc += ((aboneler+i)->kurslar + j)->aidat*((aboneler+i)->kurslar + j)->sure;

        }
        (aboneler+i)->toplamAidat = tBorc;
        printf("Abone no: %d, ToplamAidat: %d \n",i+1,(aboneler+i)->toplamAidat);
    }
}
```