Question 1)

#include Struct friends { Char name[32]; Char pet[32]; Int pn; }; Struct type { Char tof[20]; Char nocf[20]; Char pvt[20]; }; Void main() { int n; Printf(“\nEnter the number of friends:”); Scanf(“%d”,&n); Struct friends fnd[n];//friend Struct type tp[n];//type For (int i = 0 ; i < n ; i++) Printf(“\nEnter Friend name:”); Scanf(“%s”,&fnd[i].name); Printf(“\nEnter Pet name:”); Scanf(“%s”,&fnd[i].pet); Printf(“\nEnter phone number:”); Scanf(“%d”,&fnd[i].pn); Printf(“\nEnter the type of friend:”); Scanf(“%s”,&tp[i].tof); Printf(“\nEnter the Name of common friends:”); Scanf(“%s”,&tp[i].nocf); Printf(“\nEnter the places visited together:”); Scanf(“%s”,&tp[i].pvt); } Printf(“\nFriends list”); For(int i = 0 ; i < n ; i++) { Printf(“\n\nS.No:%d”,i+1); Printf(“\nName:%s”,fnd[i].name); Printf(“\nPet name:%s”,fnd[i].pet); Printf(“\nPhone number:%d”,fnd[i].pn); Printf(“\nType of friend:%s”,tp[i].tof); Printf(“\nName of common friends:%s”,tp[i].nocf); Printf(“\nPlaces visited together:%s”,tp[i].pvt); } }

Question 2)

#include #include Struct product { Char name[50]; Char Id[50]; Float price; }; Int main() { Struct product \*pdt; Int n; Printf(“\nEnter the number of product:”); Scanf(“%d”,&n); Pdt = (struct product \*)malloc(n \* sizeof(struct product)); For(int i = 0;i < n ; i++ ) { Printf(“\nEnter details of product %d”,i+1); Printf(“\n\nProduct Name:”); Scanf(“%s”,&(pdt+i)->name); Printf(“\nProduct Id:”); Scanf(“%s”,&(pdt+i)->Id); Printf(“\nPrice:”); Scanf(“%f”,&(pdt+i)->price); } For (int i = 0 ; i < n ; i++) { Printf(“\n\nProduct details:”); Printf(“\nProduct Name:%s,\nProduct ID:%s,\nProduct price:%f”,(pdt+i)->name,(pdt+i) >Id,(pdt+i)->price); } Float totalCost = 0; For (int i = 0; i < n; i++) { totalCost += (pdt + i)->price; } Printf(“\n\nTotal cost for all products: %f\n”, totalCost); Float maxprice = pdt->price; Float minprice = pdt->price; Int maxi = 0, mini = 0; //maxi = maxindex For (int i = 1; i < n; i++) { If ((pdt + i)->price > maxprice) { Maxprice = (pdt + i)->price; Maxi = i; } If ((pdt + i)->price < minprice) { Minprice = (pdt + i)->price; Mini = i; } } Printf(“\n\nMost expensive product:\n”); Printf(“Name: %s, ID: %s, Price: %f\n”, (pdt + maxi)->name, (pdt + maxi)->Id, maxprice); Printf(“\nLeast expensive product:\n”); Printf(“Name: %s, ID: %s, Price: %f\n”, (pdt + mini)->name, (pdt + mini)->Id, minprice); Free(pdt); Return 0; }