**C Program for Segmentation.**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct list

{

int seg;

int base;

int limit;

struct list \*next;

} \*p;

void insert(struct list \*q,int base,int limit,int seg)

{

if(p==NULL)

{

p=malloc(sizeof(struct list));

p->limit=limit;

p->base=base;

p->seg=seg;

p->next=NULL;

}

else

{

while(q->next!=NULL)

{

q=q->next;

printf("yes");

}

q->next=malloc(sizeof(struct list));

q->next ->limit=limit;

q->next ->base=base;

q->next ->seg=seg;

q->next ->next=NULL;

}

}

int find(struct list \*q,int seg)

{

while(q->seg!=seg)

{

q=q->next;

}

return q->limit;

}

int search(struct list \*q,int seg)

{

while(q->seg!=seg)

{

q=q->next;

}

return q->base;

}

int main()

{

p=NULL;

int seg,offset,limit,base,c,s,physical;

printf("Enter segment table\n");

printf("Enter -1 as segment value for termination\n");

do

{

printf("Enter segment number");

scanf("%d",&seg);

if(seg!=-1)

{

printf("Enter base value:");

scanf("%d",&base);

printf("Enter value for limit:");

scanf("%d",&limit);

insert(p,base,limit,seg);

}

}

while(seg!=-1);

printf("Enter offset:");

scanf("%d",&offset);

printf("Enter bsegmentation number:");

scanf("%d",&seg);

c=find(p,seg);

s=search(p,seg);

if(offset<c)

{

physical=s+offset;

printf("Address in physical memory %d\n",physical);

}

else

{

printf("error");

}

return 0;

}