# TASK 1:

**PSEUDOCODE:**

Struct page {page, frame};

Input=> logical address, physical address, size of page;

No of frames = physical address / size of page;

No of pages = logical address / size of page;

Initialize all pages = -1;

Initialize page no = frame No;

Check for duplicate;

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

struct page {

int pages;

int frames;

};

int main()

{

int log\_add, phy\_add, page\_size;

printf("Enter size of logical address: ");

scanf("%d", &log\_add);

printf("Enter size of physical address: ");

scanf("%d", &phy\_add);

printf("Enter size of pages: ");

scanf("%d", &page\_size);

int no\_pages, no\_frames;

no\_pages = log\_add / page\_size;

no\_frames = phy\_add / page\_size;

int frame\_table[no\_frames], temp;

struct page obj[no\_pages];

for(int i=0; i<no\_pages; i++)

{

obj[i].pages = i;

obj[i].frames = -1;

}

for(int i=0; i<no\_frames; i++)

{

frame\_table[i] = 0;

}

srand(time(NULL));

for(int i=0; i<no\_pages; i++)

{

temp = rand() % no\_frames;

while(frame\_table[temp] == 1)

{

temp = rand() % no\_frames;

}

frame\_table[temp] = 1;

obj[i].frames = temp;

}

int logical\_addresses[no\_pages];

for(int i=0; i<no\_pages; i++)

{

printf("Enter logical address of page %d: ", i);

scanf("%d", &logical\_addresses[i]);

}

for(int i=0; i<no\_pages; i++)

{

int base = obj[i].frames \* page\_size;

int offset = logical\_addresses[i] % page\_size;

int physical\_address = base + offset;

printf("\nPage No: %d Frame No: %d Base address: %d Physical address: %d", obj[i].pages, obj[i].frames, base, physical\_address);

}

return 0;

}

# OUTPUT:

