#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

void\* runner(void \*parameters)

{int\* pp=(int\*)calloc(20, sizeof(int));

int i;

int vv;

pp[1]=0;

pp[2]=1;

vv=3;

for(i=3; ; i++)

{

pp[i]=pp[i-1]+pp[i-2];

vv++;

if(pp[i]>=(int)parameters)

{

pp[0]=vv;

break;

}

}

printf("Thread: %d\n", (int)parameters);

return pp;

pthread\_exit(0);

}//End runner

int main(int argc, char\*argv[])

{

//thread identifier

pthread\_t threadID[argc-1];

//set attributes for the thread

pthread\_attr\_t attributes;

//get the default attributes

pthread\_attr\_init(&attributes);

//create the thread

int y=0, z=0, v=0;

for(y=0; y<(argc-1); y++)

{

printf("-----%d\n", atoi(argv[y+1]));

pthread\_create(&threadID[y], &attributes, runner,(atoi(argv[y+1])));

}

//now wait for the thread to exit

for(z=0; z<(argc-1); z++)

{ int\* result;

pthread\_join(threadID[z], &result);

printf("%d: ",atoi(argv[z+1]));

for(v=1;v<result[0] ; v++)

{

printf("%d ", result[v]);

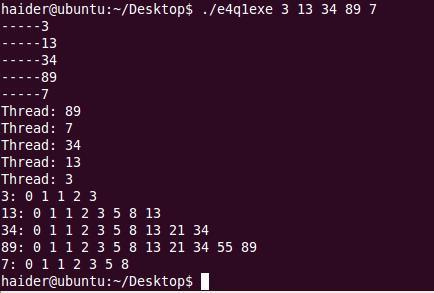
}

free(result);

printf("\n");

}

}



Q#2

#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

int siz;

//this data is shared by the thread(s)

//The thread will begin control in this funtion

void\* runner1(char\* argv[])

{

int a;

int b=0;

for(a=1;a<siz;a++)

b+=atoi(argv[a]);

pthread\_exit((void\*)(b/(siz-1)));

}

void\* runner2(char\* argv[])

{

int a;

int b=atoi(argv[1]);

for(a=2;a<siz;a++)

if(b>atoi(argv[a]))

b=atoi(argv[a]);

pthread\_exit((void\*)b);

}

void\* runner3(char\* argv[])

{

int a;

int b=0;

for(a=1;a<siz;a++)

if(b<atoi(argv[a]))

b=atoi(argv[a]);

pthread\_exit((void\*)b);

}

int main(int argc, char\*argv[])

{

int i;

int mn,mx;

siz=argc;

//thread identifier

pthread\_t threadID1,threadID2,threadID3;

//set attributes for the thread

pthread\_attr\_t attributes;

//get the default attributes

pthread\_attr\_init(&attributes);

//create the thread

pthread\_create(&threadID1, &attributes, runner1, argv);

pthread\_join(threadID1, (void\*)&i);

printf("Average: %d\n",i);

pthread\_create(&threadID2, &attributes, runner2, argv);

pthread\_join(threadID2, (void\*)&mn);

printf("Minimum: %d\n",mn);

pthread\_create(&threadID3, &attributes, runner3, argv);

pthread\_join(threadID3, (void\*)&mx);

printf("Maximum: %d\n",mx);

}