PI code

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

int main(int argc, char **argv)
{
  clock_t begin = clock();
  char *str = argv[1];
  char *e;
  long STEPS = atoi(str);
  double x, y, z, pi;
  long count = 0;
  for (int i = 0; i <= STEPS; i++)
  {
    x = rand()/(double)RAND_MAX;
    y = rand()/(double)RAND_MAX;
    z = x*x + y*y;
    if (z <= 1)
      count++;
  }
  pi = (double)count/STEPS*4;
  // printf("N = %lu\t", STEPS);
  // printf("Pi = %.20lf\n", pi);

  clock_t end = clock();
  double time = (long double)(end - begin) / CLOCKS_PER_SEC;

  FILE *file = fopen("g.txt", "a+");
  fprintf(file, "%lu %lf %lf\n", STEPS,time,pi);
  fclose(file);
  return 0;
}
```

Shell Code

```
#!/bin/bash

i=1

gcc pi.c

while [ $i -le 100000 ]

  do
```

```
    ./a.out $i
    i=`expr $i + 200`
  done
awk 'function modulus(x) {if(x < 0) return x*=-1; else return x;}{print $0,modulus($4=$3-3.14159265358979311600)}' g.txt > temp.txt
awk '{print $0,$5=$4/3.14159265358979311600}' temp.txt > g.txt
echo "Steps    Time    Abs-Err    Rel-Abs-Err " && awk '{print $1,"    ",$2,"    ",$4,"    ",$5,"\n"}' g.txt
gnuplot << EOF
set term wxt enhanced
set terminal png size 2000,1000
set title 'Graph'
set output 'plot1.png'
set xlabel 'STEPS'
set ylabel 'Time'
plot 'g.txt' using 1:2 w l title 'PI'
EOF
gnuplot << EOF
set term wxt enhanced
set terminal png size 2000,1000
set title 'Graph'
set output "plot2.png"
set xlabel 'STEPS'
set ylabel 'Abs and Rel-Abs %'
set yrange [0:1]
set xrange [0:25000]
plot 'g.txt' using 1:4 lc rgb 'black' w l  title 'Abs-Error', 'g.txt' using 1:5 w l lc rgb 'red' title 'Relative-Abs-Error %'
EOF
```

```
echo "Done!"
rm g.txt
xdg-open plot1.png
xdg-open plot2.png
```