

1. W systemie pomocy znajdź opis programu do archiwizacji danych tar. Spakować wszystkie pliki znajdujące się w katalogu bieżącym (tj. /home) do pojedynczego archiwum o nazwie programy.tar (opcje -cvf). Utworzyć w katalogu domowym, katalog o nazwie Dokumenty. Skopiować archiwum do katalogu /home/Dokumenty i tam go rozpakować (opcje -xvf).

```
tar -cvf /home/* programy.tar
```

```
mkdir /home/Dokumenty
```

```
cp -R /home programy.tar /home/Dokumenty
```

```
tar -xvf /home/Dokumenty/programy.tar
```

2. Zmodyfikować program obliczający tablicę funkcji sinus, tak, aby zawierał instrukcję sterującą for. Utworzyć funkcję sinus i przenieść jej kod do oddzielnego pliku źródłowego sinus.c. Dołączyć plik nagłówkowy sinus.h z deklaracją funkcji. Napisać odpowiedni skrypt Makefile. Wyniki zapisać do pliku sinus.dat.

Pierwotnie:

```
#include <math.h>

#include <stdio.h>

int main()
{
    int angle_degree;

    double angle_radian, pi, value;

    printf ("\nOblicz tablice funkcji sinus...\n\n");

    pi = 4.0*atan(1.0);

    printf ( " wartosc PI = %f \n\n", pi );

    printf ( " kat \n" );

    angle_degree=0.0;

    while( angle_degree <= 360.0 )
    {
        angle_radian = pi * angle_degree/180.0;

        value = sin(angle_radian);

        printf(" %3d  %f \n", angle_degree, value);

        angle_degree = angle_degree + 10.0;
    }

    return 0;
}
```

Po zmianie:

```

#include <math.h>

#include <stdio.h>

int main()
{
    int angle_degree;

    double angle_radian, pi, value;

    FILE *fp;

    fp = fopen ("sinus.dat", "w");

    printf ("\nOblicz tablice funkcji sinus...\n\n");

    pi = 4.0*atan(1.0);

    printf ( " wartosc PI = %f \n\n", pi );

    printf ( " kat \n" );

    angle_degree=0.0;

    for(angle_degree = 0; angle_degree <= 360; angle_degree = angle_degree + 10)
    {

        angle_radian = pi * angle_degree/180.0;

        value = sin(angle_radian);

        printf(" %3d  %f \n", angle_degree, value);

        fprintf(fp, " %3d  %f \n", angle_degree, value);

    }

    fclose (fp);

    return 0;

}

```

3. *Napisz funkcję sprawdzającą ile par liczb całkowitych z przedziału $\langle a, b \rangle$ jest zawartych w kole o średnicy 8. ($x^2 + y^2 \leq 8$). Wartości a i b powinny być zadawane z klawiatury i przekazywane jako parametry funkcji.*

```

#include <math.h>

#include <stdio.h>

int main()
{

    int a, b, x, y, var;

```

```

int counter = 0;

printf("Podaj a i b: ");

scanf("%d %d", &a, &b);

printf("a = %d b = %d", a, b);

if(a>b)
{
    printf("\na musi byc mniejsze od b!");
}
else
{
    x = a;
    y = a;
    while(x<=b)
    {
        while(y<=b)
        {
            var = x^2+y^2;
            if(var<=8)
            {
                counter++;
            }
            y++;
        }
        y = a;
        x++;
    }

    printf("\nLiczba par liczb calkowitych z przedzialu <a, b> zawartych w kole jest rowna:
%d", counter);
}

return 0;
}

```

4. Napisać program, w którym proces macierzysty P tworzy dwa procesy potomne $R1$, $R2$. Każdy z procesów potomnych powinien utworzyć swój własny proces potomny: $R1 \rightarrow S1$, $R2 \rightarrow S2$. Każdy z procesów (z wyjątkiem P) niech wyświetli swój PID oraz PPID.

