**Эта программа реализует задание в л.р.2 и должна быть немного трансформирована, исходя из вашего варианта и пояснений, представленных в учебном пособии**

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% File Nelson.m.-\

% Выполняет запуск модели ПО , созданной в среде MatLab Simulink

% c не обнаруженными ошибками и выполняет обработку согласно

% моделей надёжности Nelson C.E. и Corcoran W.J.

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear;

npusk=10; % Число функциональных разрезов

Stop\_Time=100.0; % время "выполнения" имитационной модели ПО

vector\_osh=zeros(1,npusk);

for k=1 : npusk;

sim( 'PO\_imit');% Пуск модели

vector\_osh(k)=mean(osybky.signals.values); % Среднее число ошибок

end

p\_otk=vector\_osh/Stop\_Time;

figure(1);

stem(1:npusk,p\_otk,'k');

set(gca,'FontName','Arial Cyr','FontSize',8)

title('Распределение вероятностей отказа по функциональным разрезам');

xlabel('№ разреза');

ylabel('\it P \_{Отк j}');

grid;

r\_n=cumprod(1.0-p\_otk); % показатель надёжности по разрезам

figure(2);

stem(1:npusk,r\_n,'MarkerFaceColor','black','Marker','diamond');

set(gca,'FontName','Arial Cyr','FontSize',8)

title('Оценки вероятностей безотказной работы');

xlabel('№ испытания');

ylabel('\it P');

grid;

r\_s=sum(p\_otk); % показатель степени экспоненты

% Оценка Corcoran W.J.

isp=npusk\*Stop\_Time; % число испытаний

r\_c=(isp-sum(vector\_osh)+sum(vector\_osh-1))/isp; % коэффициент

figure(3);

axis('off');

h1=text(0.3,0.9,'РЕЗУЛЬТАТЫ','FontSize', 20);

h1=text(0.0,0.7,sprintf('Число функциональных разрезов %g',npusk),'FontSize', 12);

h1=text(0.0,0.65,sprintf('Число испытаний %g',isp),'FontSize', 12);

h1=text(0.0,0.6,sprintf('Оценка Коркорэна %g',r\_c),'FontSize', 12);

h1=text(0.0,0.35,sprintf('Оценка надёжности Нельсона %g',r\_s),'FontSize', 12);