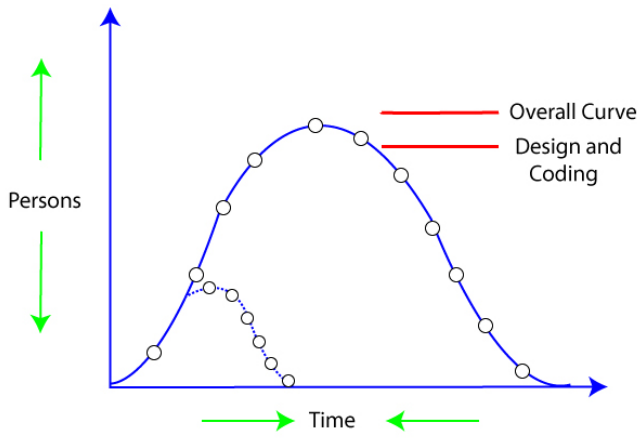
SOFTWARE PROJECT MANAGEMENT LAB - G2
EXPERIMENT 8

- ASHISH KUMAR
- 2K18/SE/041

AIM:- Write a program to implement **Putnam resource allocation model** and calculate peak manning and average rate of software team build up.

THEORY:-

The **Lawrence Putnam model** describes the time and effort requires finishing a software project of a specified size. Putnam makes a use of a so-called The **Norden/Rayleigh Curve** to estimate project effort, schedule & defect rate as shown in figure:



The Rayleigh manpower loading Curve

Putnam noticed that software staffing profiles followed the well known Rayleigh distribution. Putnam used his observation about productivity levels to derive the **software equation**:

$$L = C_k K^{1/3} t_d^{4/3}$$

The various terms of this expression are as follows:

K is the total effort expended (in PM) in product development, and L is the product estimate in KLOC. t_d correlate to the time of system and integration testing. Therefore, t_d can be relatively

considered as the time required for developing the product. C_k is the state of technology constant and reflects requirements that impede the development of the program.

Typical values of $\mathbf{C_k = 2}$ for poor development environment

$\mathbf{C_k = 8}$ for good software development environment

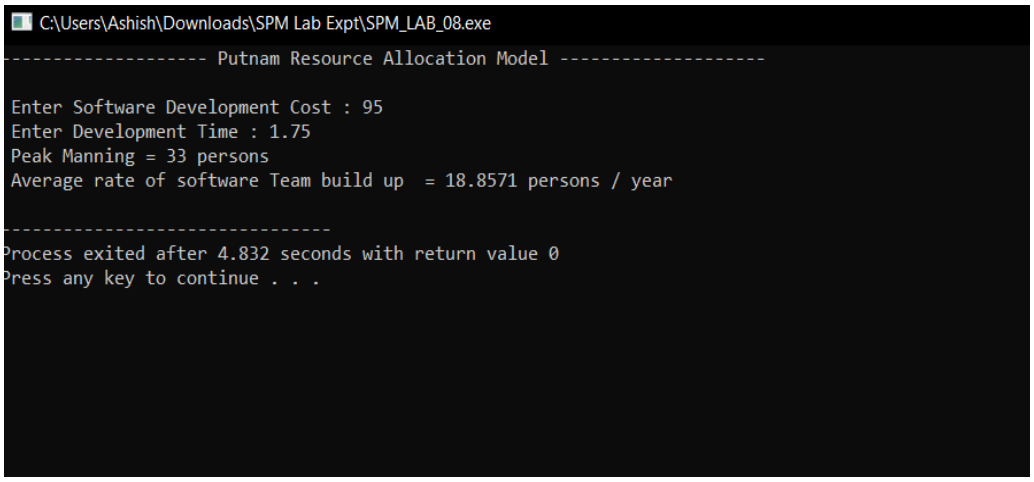
$\mathbf{C_k = 11}$ for an excellent environment (in addition to following software engineering principles, automated tools and techniques are used).

Putnam proposed that optimal staff develop on a project should follow the Rayleigh curve. Only a small number of engineers are required at the beginning of a plan to carry out planning and specification tasks. As the project progresses and more detailed work are necessary, the number of engineers reaches a peak. After implementation and unit testing, the number of project staff falls.

CODE:-

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;

int main(){
// Given the Software Development Cost and the Peak Development Time
int K;
cout<<"---------- Putnam Resource Allocation Model ----------\n\n";
double T;
cout << " Enter Software Development Cost : ";
cin >> K;
cout << " Enter Development Time : ";
cin >> T;

double peakmanning = K / (T * 1.64);
cout <<" Peak Manning = "<< round(peakmanning) <<" persons "<< endl;

double avg_team_buildup_rate = round(peakmanning) / T;
cout <<" Average rate of software Team build up  = " << avg_team_buildup_rate <<" persons /
year "<< endl;

return 0;
}
```

OUTPUT:-



Findings & Learning: -

- We have successfully implemented Putnam resource allocation model and calculated peak manning and average rate of software team build up.
- Putnam makes a sharp distinction between 'conventional productivity': size / effort and process productivity.
- One of the key advantages to this model is the simplicity with which it is calibrated. Most software organizations, regardless of maturity level can easily collect size, effort and duration (time) for past projects. Process Productivity, being exponential in nature is typically converted to a linear productivity index an organization can use to track there own changes in productivity.