**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **SHELL SCRIPT PROGRAMS:-** | **Pg No** |
| **1** | **To Demonstrate a simple Unix shell program** | **3** |
| **2** | **Shell Script Program to Print Student Mark Sheet** | **4** |
| **3** | **Shell Script Program to perform all Arithmetic operations on integers** | **5** |
| **4** | **Shell Script Program to perform all Arithmetic operations on floating point** | **6** |
| **5** | **Shell Script Program to find simple interest** | **7** |
| **6** | **Shell Script Program to find Area of Square, Rectangle, Circle** | **8** |
| **7** | **Shell Script Program to print your Address ‘n’ times** | **9** |
| **8** | **Shell Script Program to find whether number is even or odd** | **10** |
| **9** | **Shell Script Program to find whether number is +ve, -ve or 0** | **11** |
| **10** | **Shell Script Program to find Greatest of 3 numbers** | **12** |
| **11** | **Shell Script Program to whether year is Leap year or not** | **13** |
| **12** | **Shell Script Program to check whether the given number is divisible by 11 or not** | **14** |
| **13** | **Shell Script Program to print natural numbers from 1 to 10 using WHILE loop** | **15** |
| **14** | **Shell Script Program to print perfect numbers from 1 to 100** | **16** |
| **15** | **Shell Script Program to reverse a number** | **17** |
| **16** | **Shell Script Program to find whether the given number is perfect or not** | **18** |
| **17** | **Shell Script Program to find sum of digits of a number** | **19** |
| **18** | **Shell Script Program to print multiplication table of any number using FOR loop** | **20** |
| **19** | **Shell Script Program to print Prime numbers from 1 to 20** | **21** |
| **20** | **Shell Script Program to find whether the given number is Prime or not** | **22** |
| **21** | **Shell Script Program to check whether the given number is Armstrong or not** | **23** |
| **22** | **Shell Script Program to print Armstrong nos from 1-1000** | **24** |
| **23** | **Shell Script Program to find Factorial of a number** | **25** |
| **24** | **Shell Script Program to print the Fibonacci series** | **26** |
| **25** | **Shell Script Program to print Employee Payroll** | **27** |
| **26** | **Shell Script Program to implement Break statement** | **28** |
| **27** | **Shell Script Program to implement Continue statement** | **29** |
|  | **Shell Script Program to Input the day number from the keyboard and print the corresponding day** | **30** |
| **28** | **Using else if ladder** | **32** |
| **29** | **Using elif structure** | **33** |
| **30** | **Using case control structure** | **34** |
| **31** | **Shell script Program to accept a character and check whether it is an**   * **Lower case alphabet** * **Upper case alphabet** * **A digit** * **Special symbol** * **Vowel**   **Using case control structure** | **35** |
| **32** | **Shell Script Program to count number of odd and even digits within a given number** | **36** |
| **33** | **Using case .. esac structure**   * **Find the number of users logged into the system** * **Print the calendar for current year** * **Print the date** | **37** |
| **34** | **Shell Script Program to perform all Arithmetic Operations using Command line arguments** | **38** |
| **35** | **Shell Script Program to check whether given file is a directory or not** | **39** |
| **36** | **Shell Script Program to Count number of files in a Directory** | **40** |
| **37** | **Shell Script Program to search whether element is present is in the list or not** | **42** |
| **38** | **Shell Script Program to implement read, write, execute permissions** | **43** |
| **39** | **Shell Script Program to copy contents of one file to another** | **44** |
| **40** | **Shell Script Program for FCFS Algorithm** | **44** |
| **41** | **Shell Script Program for SJF Algorithm** | **46** |
| **42** | **Shell Script Program for Priority Scheduling** | **48** |

**#1. To Demonstrate a simple Unix shell program**

**-->$ vi welcome.sh**

**write the code below in it**

**-----------------------------------------------------------------------------------------**

**echo "Welcome to Unix"**

**echo “Enter X”**

**read x**

**echo “x = “ $x**

**------------------------------------------------------------------------------------------**

**then press "esc + :wq!" to save it**

**-->$ sh welcome.sh**

**------------------------------------------------------------------------------------------**

**Output :**

**Welcome to Unix**

**Enter X**

**34**

**X = 34**

**#2. To Print Student Mark Sheet**

**echo "Enter 3 Students Names & Marks : "**

**read n1**

**read m1**

**read n2**

**read m2**

**read n3**

**read m3**

**echo "Name" " " "Marks"**

**echo $n1 " " $m1**

**echo $n2 " " $m2**

**echo $n3 " " $m3**

**Output :**

**Enter 3 Students Names & Marks :**

**satya**

**09**

**uday**

**99**

**satish**

**40**

**Name Marks**

**satya 09**

**uday 99**

**satish 40**

**#3. To perform all Arithmetic operations on integers**

**echo "enter the first no."**

**read a**

**echo "enter the 2nd no."**

**read b**

**c=`expr $a + $b`**

**echo "sum : "$c**

**c=`expr $a - $b`**

**echo "diff : "$c**

**c=`expr $a \\* $b`**

**echo "mul : "$c**

**c=`expr $a / $b`**

**echo "div : "$c**

**c=`expr $a % $b`**

**echo "modulus : "$c**

**Output :**

**enter the first no.**

**36**

**enter the 2nd no.**

**34**

**sum : 70**

**diff : 2**

**div : 1**

**mul : 1224**

**modulus : 2**

**#4. To perform all Arithmetic operations on floating point**

**echo "Enter the value of a"**

**read a**

**echo "Enter the value of b"**

**read b**

**c=`echo $a + $b|bc`**

**echo "sum : "$c**

**c=`echo $a - $b|bc`**

**echo "diff : "$c**

**c=`echo $a \\* $b|bc`**

**echo "mul : "$c**

**c=`echo $a / $b|bc`**

**echo "div : "$c**

**c=`echo $a % $b|bc`**

**echo "modulus : "$c**

**Output :**

**Enter the value of a**

**3.7**

**Enter the value of b**

**3.4**

**sum : 7.1**

**diff : 0.3**

**mul : 12.58**

**div : 1**

**modulus : 0.3**

**#5. To find simple interest**

**clear**

**echo "Enter P,N and R"**

**read p**

**read n**

**read r**

**si=`expr \( $p \\* $n \\* $r \) / 100`**

**echo "Simple Intrest is " si**

**Output :**

**Enter P,N and R**

**1000**

**5**

**2**

**Simple Intrest is 100**

**#6. To find area of circle, Rectangle and Square**

**echo “Enter radius”**

**read r**

**echo "Enter Length & Breadth"**

**read l**

**read b**

**echo "Enter side"**

**read s**

**a=`echo $r \\* $r \\* 3.14|bc`**

**echo “Area of Circle ” $a**

**a=`echo $l \\* $b`**

**echo “Area of Rectangle ” $a**

**a=`echo $s \\* $s`**

**echo “Area of Square ” $a**

**OUTPUT**

**Enter radius**

**5**

**Enter Length & Breadth**

**20**

**10**

**Enter side**

**20**

**Area of Circle 78.5**

**Area of Rectangle 200**

**Area of Square 400**

**#7. To print your Address ‘n’ times**

**echo "Enter n : "**

**read n**

**echo "Enter Your Plot No"**

**read plot**

**echo "Enter u r Street Name"**

**read street**

**echo "Enter u r city"**

**read city**

**echo "Printing Address " $n " times :"**

**for ((i=0;i<n;i++))**

**do**

**echo $plot " , " $street " , " $city**

**done**

**Output :**

**Enter n :**

**3**

**Enter Your Plot No**

**34-3637/NCS**

**Enter u r Street Name**

**Suraram**

**Enter u r city**

**Hyderabad**

**Printing Address 3 times :**

**34-3637/NCS , Suraram , Hyderabad**

**34-3637/NCS , Suraram , Hyderabad**

**34-3637/NCS , Suraram , Hyderabad**

**#8. To find whether the given number is even or odd**

**clear**

**echo "Enter n"**

**read n**

**if [ `expr $n % 2` -eq 0 ]**

**then**

**echo "n is even"**

**else**

**echo "n is odd"**

**fi**

**Output :**

**Enter n**

**34**

**n is even**

**Enter n**

**37**

**n is odd**

**#9. To find whether number is positive,negative or zero**

**clear**

**echo "Enter n"**

**read n**

**if [ $n -gt 0 ]**

**then**

**echo "n is positive"**

**else**

**if [ $n -eq 0 ]**

**then**

**echo "n is zero"**

**else**

**echo "n is negative"**

**fi**

**fi**

**Output :**

**Enter n**

**34**

**n is positive**

**#10. To find Greatest of 3 numbers**

**clear**

**echo "Enter any 3 numbers"**

**read x**

**read y**

**read z**

**if [ $x -gt $y -a $x -gt $z ]**

**then**

**echo $x " is Great"**

**else**

**if [ $y -gt $z ]**

**then**

**echo $y " is Great"**

**else**

**echo $z " is Great"**

**fi**

**fi**

**Output :**

**Enter any 3 numbers**

**36**

**37**

**34**

**37 is Great**

**#11. To find whether year is leap year or not**

**echo "Enter the year"**

**read y**

**r=`expr $y % 4`**

**if [ $r -eq 0 ]**

**then**

**echo $r " is a Leap Year"**

**else**

**echo $r " is not a Leap Year"**

**fi**

**Output :**

**Enter the year**

**2004**

**2004 is a Leap Year**

**#12. To check whether number is divisible by 11 or not**

**echo "Enter any Number"**

**read n**

**r=`expr $y % 11`**

**if [ $r -eq 0 ]**

**then**

**echo $r " is divisible by 11"**

**else**

**echo $r " is not divisible by 11"**

**fi**

**Output :**

**Enter any Number**

**121**

**121 is divisible by 11**

**#13. To print natural numbers from 1 to 10 using WHILE loop**

**echo "1-10 Natural No's are :"**

**i=1**

**while [ $i -le 10 ]**

**do**

**echo $i**

**i=`expr $i + 1`**

**done**

**Output:**

**1-10 Natural No's are :**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**#14. To print perfect numbers from 1 to 100**

**echo "Perfect Numbers 1 - 100 are : "**

**n=1**

**while [ $n -lt 100 ]**

**do**

**x=$n**

**sum=0**

**for ((i=1;i<x;i++))**

**do**

**r=`expr $x % $i`**

**if [ $r = 0 ]**

**then**

**sum=`expr $sum + $i`**

**fi**

**done**

**if [ $sum -eq $n ]**

**then**

**echo $n**

**fi**

**n=`expr $n + 1`**

**done**

**Output :**

**Perfect Numbers 1 - 100 are :**

**6**

**28**

**#15. To reverse a number**

**clear**

**echo "Enter a Number"**

**read n**

**s=0**

**while [ $n -gt 0 ]**

**do**

**r=`expr $n % 10`**

**s=`expr $s \\* 10 + $r`**

**n=`expr $n / 10`**

**done**

**echo "Reverse of it is" $s**

**Output:**

**Enter a Number**

**123**

**Reverse of it is 321**

**#16. To find whether number is perfect or not**

**echo "Enter a no"**

**read n**

**sum=0**

**for ((i=1;i<n;i++))**

**do**

**r=`expr $n % $i`**

**if [ $r = 0 ]**

**then**

**sum=`expr $sum + $i`**

**fi**

**done**

**if [ $sum -eq $n ]**

**then**

**echo $n " is perfect"**

**else**

**echo $n " is not perfect"**

**fi**

**Output :**

**Enter a no**

**28**

**28 is perfect**

**Enter a no**

**34**

**34 is not perfect**

**#17. To find sum of digits of a number**

**echo "Enter the number"**

**read n**

**sum=0**

**while [ $n -gt 0 ]**

**do**

**a=`expr $n % 10`**

**sum=`expr $sum + $a`**

**n=`expr $n / 10`**

**done**

**echo "The sum of digits of no. is : "$sum**

**Output:**

**Enter the number**

**1234**

**The sum of digits of no. is : 10**

**#18. To print multiplication table using FOR loop**

**echo "Enter n"**

**read n**

**echo "Multiplication Table"**

**for ((i=1;i<10;i++))**

**do**

**echo $n " \* " $i " = " `expr $n \\* $i`**

**i=`expr $i + 1`**

**done**

**Output :**

**Enter n**

**34**

**Multiplication Table**

**34 \* 1 = 34**

**34 \* 2 = 68**

**34 \* 3 = 102**

**34 \* 4 = 136**

**34 \* 5 = 170**

**34 \* 6 = 204**

**34 \* 7 = 238**

**34 \* 8 = 272**

**34 \* 9 = 306**

**34 \* 10 = 340**

**#19. To print prime numbers 1-20**

**echo "Prime numbers 1-20 are : "**

**j=1**

**while [ $j -lt 20 ]**

**do**

**i=1**

**c=0**

**while [ $i -lt $j ]**

**do**

**r=`expr $n % $i`**

**if [ $r -eq 0 ]**

**then**

**c=`expr $c + 1`**

**fi**

**i=`expr $i + 1`**

**done**

**if [ $c -eq 0 ]**

**then**

**echo $j**

**fi**

**j=`expr $j + 1`**

**done**

**Output :**

**2**

**3**

**5**

**7**

**11**

**13**

**17**

**19**

**#20. To find whether number is prime or not**

**echo "Enter n"**

**read n**

**i=1**

**c=0**

**while [ $n -gt $i ]**

**do**

**r=`expr $n % $i`**

**if [ $r -eq 0 ]**

**then**

**c=`expr $c + 1`**

**fi**

**i=`expr $i + 1`**

**done**

**if [ $c -eq 0 ]**

**then**

**echo "It is a prime"**

**else**

**echo "It is not a prime"**

**fi**

**Output :**

**Enter n**

**34**

**It is not a prime**

**#21. To check whether number is Armstrong or not**

**echo "Enter a no"**

**read n**

**sum=0**

**r=0**

**x=$n**

**while [ $x -gt 0 ]**

**do**

**r=`expr $n % 10`**

**sum=`expr $sum + $r \\* $r \\*r`**

**x=`expr $n / 10`**

**done**

**if [ $n -eq $sum ]**

**then**

**echo $n " is Armstrong"**

**else**

**echo $n " is not Armstrong"**

**fi**

**Output :**

**Enter a no**

**153**

**153 is Armstrong**

**#22. To print Armstrong numbers from 1 to 1000**

**echo "Armstrong Numbers 1 - 100 are : "**

**n=1**

**while [ $n -lt 1000 ]**

**do**

**x=$n**

**sum=0**

**while [ $x -gt 0 ]**

**do**

**r=`expr $x % 10`**

**sum=`expr $sum + $r \\* $r \\* $r`**

**x=`expr $x / 10`**

**done**

**if [ $sum -eq $n ]**

**then**

**echo $n**

**fi**

**n=`expr $n + 1`**

**done**

**Output :**

**1**

**153**

**370**

**371**

**407**

**#23. To find Factorial of a number**

**clear**

**echo "Enter n"**

**read n**

**f=1**

**while [ $n -gt 0 ]**

**do**

**f=`expr $f \\* $n`**

**n=`expr $n - 1`**

**done**

**echo "Factorial is " $f**

**Output :**

**Enter n**

**5**

**Factorial is 120**

**#24. To print the Fibonacci series**

**echo "Enter limit :"**

**read n**

**f1=0**

**f2=1**

**echo "The Fibonacci sequence is : "**

**for (( i=0;i<=n;i++ ))**

**do**

**echo $f1**

**temp=$f2**

**f2=`expr $f1 + $f2`**

**f1=$temp**

**done**

**Output :**

**Enter limit :**

**8**

**The Fibonacci sequence is :**

**0**

**1**

**1**

**2**

**3**

**5**

**8**

**13**

**#25. To print Employee Payroll**

**echo “Enter Basic Salary”**

**read basic**

**da=`echo 0.1 \\* $basic|bc`**

**hra=`echo 1.2 \\* $basic|bc`**

**pf=`echo 0.8 \\* $basic|bc`**

**gross=`echo $basic + $da + $hra|bc`**

**net=`echo $gross - $pf|bc`**

**echo “DA is” $da**

**echo “HRA is” $hra**

**echo “PF is” $pf**

**echo “Gross Salary is” $gross**

**echo “Net Salary is” $net**

**OUTPUT**

**Enter Basic Salary**

**100000**

**DA is 10000**

**HRA is 120000**

**PF is 80000**

**Gross Salary is 230000**

**Net Salary is 150000**

**#26. To implement Break statement**

**declare -a num[]**

**echo "Enter length of list"**

**read n**

**i=0**

**echo "Enter list of Numbers"**

**for ((i=0;i<n;i++))**

**do**

**read num[$i]**

**done**

**echo "Sum till a negative no : "**

**s=0**

**for ((i=0;i<n;i++))**

**do**

**if [ num[$i] -gt 0 ]**

**then**

**s=`expr $s + $num[$i]`**

**else**

**break**

**fi**

**done**

**echo $s**

**Output :**

**Enter length of list**

**5**

**Enter list of Numbers**

**10**

**40**

**-34**

**50**

**-36**

**Sum till a negative no :**

**50**

**#27. To implement Continue statement**

**declare -a num[]**

**echo "Enter length of list"**

**read n**

**i=0**

**echo "Enter list of Numbers"**

**for ((i=0;i<n;i++))**

**do**

**read num[$i]**

**done**

**echo "Sum of all positive nos : "**

**s=0**

**for ((i=0;i<n;i++))**

**do**

**if [ num[$i] -gt 0 ]**

**then**

**s=`expr $s + $num[$i]`**

**else**

**continue**

**fi**

**done**

**echo $s**

**Output :**

**Enter length of list**

**5**

**Enter list of Numbers**

**10**

**40**

**-34**

**50**

**-36**

**Sum of all positive nos :**

**100**

**#28. To print the corresponding day Using else if ladder**

**echo “Enter day no.”**

**read d**

**if [ $d –eq 1 ]**

**then**

**echo “Sunday”**

**else**

**if [ $d –eq 2 ]**

**then**

**echo “Monday”**

**else**

**if [ $d –eq 3 ]**

**then**

**echo “Tuesday”**

**else**

**if [ $d –eq 4 ]**

**then**

**echo “Wednesday”**

**else**

**if [ $d –eq 5 ]**

**then**

**echo “Thursday”**

**else**

**if [ $d –eq 6 ]**

**then**

**echo “Friday”**

**else**

**if [ $d –eq 7 ]**

**then**

**echo “Saturday”**

**else**

**echo “Enter correct day no. between 1-7”**

**fi**

**fi**

**fi**

**fi**

**fi**

**fi**

**fi**

**OUTPUT**

**Enter day no. 2**

**Tuesday**

**#29. To print the corresponding day Using elif structure**

**echo “Enter day no.”**

**read d**

**if [ $d –eq 1 ]**

**then**

**echo “Sunday”**

**elif [ $d –eq 2 ]**

**then**

**echo “Monday”**

**elif [ $d –eq 3 ]**

**then**

**echo “Tuesday”**

**elif [ $d –eq 4 ]**

**then**

**echo “Wednesday”**

**elif [ $d –eq 5 ]**

**then**

**echo “Thursday”**

**elif [ $d –eq 6 ]**

**then**

**echo “Friday”**

**elif [ $d –eq 7 ]**

**then**

**echo “Saturday”**

**else**

**echo “Enter the day no. between 1-7”**

**fi**

**OUTPUT**

**Enter day no. 2**

**Tuesday**

**#30. To print the corresponding day Using case control structure**

**echo “Enter day no.”**

**read d**

**case $d in**

**1)echo “Sunday”**

**;;**

**2)echo “Monday”**

**;;**

**3)echo “Tuesday”**

**;;**

**4)echo “Wednesday”**

**;;**

**5)echo “Thursday”**

**;;**

**6)echo “Friday”**

**;;**

**7)echo “Saturday”**

**;;**

**\*)echo “Enter the day no. between 1-7”**

**esac**

**OUTPUT**

**Enter day no. 2**

**Tuesday**

**#31. To accept a character and check it**

**echo “Enter any character”**

**read d**

**case $d in**

**[a,e,i,o,u,A,E,I,O,U])echo “It is a Vowel”**

**;;**

**[a-z])echo “It is a Lower case alphabet”**

**;;**

**[A-Z)echo “It is a Upper case alphabet”**

**;;**

**[0-9])echo “It is a digit”**

**;;**

**\*)echo “It is a Special Symbol”**

**esac**

**OUTPUT :**

**Enter any character**

**s**

**It is a Lower case alphabet**

**#32. To count number of odd and even digits in a number**

**echo "Enter a number"**

**read num**

**o=0**

**e=0**

**while [ $num -gt 0 ]**

**do**

**t=`expr $num % 10`**

**g=`expr $t % 2`**

**if [ $g -ne 0 ]**

**then**

**o=`expr $o + 1`**

**else**

**e=`expr $e + 1`**

**fi**

**num=`expr $num / 10`**

**done**

**echo "No of odds : "$o**

**echo "No of evens : "$e**

**Output :**

**Enter a number**

**13439**

**No of odds : 4**

**No of evens : 1**

**#33. To Use case control structure**

**ch='y'**

**while [ $ch = 'y' ]**

**do**

**echo "Enter u r choice---"**

**echo "1)No of Users Logged into System"**

**echo "2)Print Calendar for current year"**

**echo "3)Print the date"**

**echo "4)Exit"**

**read d**

**case $d in**

**1)who | wc -l**

**;;**

**2)cal 2011**

**;;**

**3)date**

**;;**

**\*)break**

**esac**

**echo "Do u wish to Continue (y/n)"**

**read ch**

**done**

**OUTPUT :**

**Enter u r choice ---**

**1)No of Users Logged into System**

**2)Print Calendar for current year**

**3)Print the date**

**4)Exit**

**1**

**5**

**Do u wish to Continue (y/n)**

**N**

**#34. To perform all Arithmetic operations using Command line argument**

**echo "The first no. is " $1**

**echo "The 2nd no. is " $2**

**c=`expr $1 + $2`**

**echo "sum : "$c**

**c=`expr $1 - $2`**

**echo "diff : "$c**

**c=`expr $1 \\* $2`**

**echo "mul : "$c**

**c=`expr $1 / $2`**

**echo "div : "$c**

**c=`expr $1 % $2`**

**echo "modulus : "$c**

**Output :**

**sh add.sh 36 34**

**The first no. is 36**

**The 2nd no. is 34**

**sum : 70**

**diff : 2**

**div : 1**

**mul : 1224**

**modulus : 2**

**#35. To check given file is a directory or not**

**echo “Enter the filename”**

**read fn**

**if [ -f $fn ]**

**then**

**echo "It is a file "**

**else**

**echo "It is a directory "**

**fi**

**output:**

**enter the filename**

**satya**

**it is a directory**

**#36. To Count number of files in a Directory**

**echo "No of files : "**

**k=0**

**for fi in \***

**do**

**k=`expr $k + 1`**

**done**

**echo $k**

**Output:-**

**No of files**

**68**

**#37. To search an element is present in list or not**

**declare -a num[]**

**echo "Enter length of list"**

**read n**

**i=0**

**echo "Enter list of Numbers"**

**for ((i=0;i<n;i++))**

**do**

**read num[$i]**

**done**

**echo "Enter element to be searched"**

**read x**

**c=0**

**for ((i=0;i<n;i++))**

**do**

**if [ $x -eq $ { num[$i] } ]**

**then**

**c=`expr $c + 1`**

**fi**

**done**

**if [ $c -gt 0 ]**

**then**

**echo "It is in the List"**

**else**

**echo "It is not in the list"**

**fi**

**Output :**

**Enter length of list**

**5**

**Enter list of Numbers**

**10**

**40**

**30**

**20**

**50**

**Enter element to be searched**

**30**

**It is in the List**

**#38. To implement read, write, execute permissions**

**echo “enter the name of file”**

**read fn**

**if [ -rwx $fn ]**

**then**

**echo "it is permitted "**

**else**

**echo "not permitted"**

**fi**

**Output:-**

**sh file1**

**it is permitted**

**#39. To copy the contents of one file to another**

**echo "Enter the source file"**

**read a**

**echo "Enter the destination file”**

**read b**

**cp $a $b**

**cat $b**

**Output:-**

**Enter the source file**

**Sounds**

**Enter the destination file**

**Noise**

**moo moo**

**bow bow**

**meow meow**

**buzz**

**blurrrr**

**moo moo**

**bow bow**

**meow meow**

**buzz**

**blurrrr**

**40. /\*cpu algorithm FCFS\*/**

**#include<stdio.h>**

**#include<conio.h>**

**void main()**

**{**

**int bt[10],wt[10],i,n,tt[10];**

**float sum,at;**

**clrscr();**

**printf("\n enter the no of process ");**

**scanf("\n %d",&n);**

**printf("\n enter the burst time of each process ");**

**for(i=0;i<n;i++)**

**{**

**printf("\n p%d ",i);**

**scanf(" %d",&bt[i]);**

**}**

**wt[0]=0;**

**for(i=1;i<n;i++)**

**{**

**wt[i]=bt[i-1]+wt[i-1];**

**sum+=wt[i];**

**}**

**printf("\n \t process name \t burst time \t waiting time \t turn around time");**

**for(i=0;i<n;i++)**

**{**

**tt[i]=bt[i]+wt[i];**

**at+=tt[i];**

**printf("\n\n\tp%d\t\t%d\t\t%d\t\t%d",i,bt[i],wt[i],tt[i]);**

**}**

**printf("\n\n\t average waiting time is %f",sum/n);**

**printf("\n\n\t average turn around time %f",at/n);**

**getch();**

**}**

**Output :**

**enter the no of process 4**

**enter the burst time of each process**

**p0 1**

**p1 4**

**p2 8**

**p3 3**

**process name burst time waiting time turn around time**

**p0 1 0 1**

**p1 4 1 5**

**p2 8 5 13**

**p3 3 13 16**

**average waiting time is 4.750000**

**average turn around time 8.750000**

**41. /\*SJF\*/**

**#include <stdio.h>**

**void main()**

**{**

**int i=0,pno[10],bt[10],n,wt[10],temp=0,j,tt[10];**

**float sum,at;**

**clrscr();**

**printf("\n enter the no of process ");**

**scanf("\n %d",&n);**

**printf("\n enter the burst time of each process");**

**for(i=0;i<n;i++,pno[i]=1){**

**printf("\n p%d",i);**

**scanf("%d",&bt[i]);**

**}**

**for(i=0;i<n;i++){**

**for(j=i+1;j<n;j++){**

**if(bt[i]>bt[j])**

**{**

**temp=bt[i];**

**bt[i]=bt[j];**

**bt[j]=temp;**

**temp=pno[i];**

**pno[i]=pno[j];**

**pno[j]=temp;**

**}**

**}**

**}**

**wt[0]=0;**

**for(i=1;i<n;i++)**

**{**

**wt[i]=bt[i-1]+wt[i-1];**

**sum=sum+wt[i];**

**}**

**printf("\n process no \t burst time\t waiting time \t turn around time\n");**

**for(i=0;i<n;i++){**

**tt[i]=bt[i]+wt[i];**

**at+=tt[i];**

**printf("\n p%d\t\t%d\t\t%d\t\t%d",i,bt[i],wt[i],tt[i]);**

**}**

**printf("\n\n\t average waiting time%f\n\taverage turn around time%f",sum/n,at/n);**

**getch();**

**}**

**Output:-**

**enter the no of process 5**

**enter the burst time of each process**

**p0 1**

**p1 5**

**p2 2**

**p3 3**

**p4 4**

**process no burst time waiting time turn around time**

**p0 1 0 1**

**p1 2 1 3**

**p2 3 3 6**

**p3 4 6 10**

**p4 5 10 15**

**average waiting time 4.000000**

**average turn around time 7.000000**

**42. /\*Priority\*/**

**#include<stdio.h>**

**void main()**

**{**

**int pno[10],i,j,bt[10],pt[10],temp=0,n,tt[10],wt[10];**

**float sum=0,at=0;**

**printf("enter the number of processes\n");**

**scanf("%d",&n);**

**for(i=0;i<n;i++) {**

**pno[i]=i+1;**

**}**

**printf("enter the burst time and priorities of each proces \n");**

**for(i=0;i<n;i++) {**

**printf("p%d \t",pno[i]);**

**scanf("%d \t %d",&bt[i],&pt[i]);**

**}**

**for(i=0;i<n;i++) {**

**for(j=i+1;j<n;j++) {**

**if(pt[i]>pt[j])**

**{**

**temp=pno[i];**

**pno[i]=pno[j];**

**pno[j]=temp;**

**temp=bt[i];**

**bt[i]=bt[j];**

**bt[j]=temp;**

**}**

**}**

**}**

**wt[0]=0;**

**for(i=1;i<n;i++) {**

**wt[i]=bt[i-1]+wt[i-1];**

**sum+=wt[i]; }**

**printf("\n \t process no \t burst time \t waiting time \t turn around time");**

**for(i=0;i<n;i++) {**

**tt[i]=bt[i]+wt[i];**

**at +=tt[i];**

**printf("\n \t p%d \t %d \t %d \t %d",pno[i],bt[i],wt[i],tt[i]);**

**}**

**printf("\n \n \t average waiting time: %f \n \t average turn around time: %f",sum/n,at/n);**

**getch();**

**}**

**Output :**

**enter the number of processes**

**4**

**enter the burst time and priorities of each proces**

**p1 5 3**

**p2 2 4**

**p3 8 1**

**p4 1 5**

**process no burst time waiting time turn around time**

**p3 8 0 8**

**p1 5 8 13**

**p2 2 13 15**

**p4 1 15 16**

**average waiting time: 9.000000**

**average turn around time: 13.000000**