questionno01. :Writeanalgorithm,drawaflowchartan

programtoconvertaBinarydecimalnumbertoitsequi

Answer:

AlgorithmtoConvertBinarytoDecimal

```
Start.
Initializedecimalnum=0 ,base=1.
ReadbinaryNum.
LoopwhilebinaryNum>0.:
remainder=binaryNum%10
decimalNum+=remainder*base
binaryNum/=10
base*=2
PrintdecimalNum.
Stop.
```

Flowchart

CProgramtoConvertBinarytoDecimal

C

```
#include<stdioh>
intmain(){
longlongbinaryNum;
intdecimalNum=0 ,remainder,base=1 ;
printf(“Enterabinarynumber:“);
scanf(“%lld“,&binaryNum);
while(binaryNum>0 ){
remainder=binaryNum%10 ;
decimalNum+=remainder* base;
binaryNum/=10 ;
base*=2;
}
printf(“Equivalentdecimalnumber:%d\n“,decimalN
return0 ;
}
```

Questionno2:Writeanalgorithmandusetheconceptof

programinC,togenerateProgressReportofstudent

forallits4 terms(theclassisof20 students)Assumpti

necessary.

Answer:

AlgorithmtoGenerateStudentProgressReport

Start.

DefineStudentstructure:studentID,name[50],ma

DeclareclassX[20]ofStudenttype.

Foreachstudent(0 to) 9):

ReadID,Name.

Foreachterm(0 to3):

InitializetotalMarks[term]=0.

Foreachsubject(0 to4):

Readmark.

AddmarktototalMarks[term].

```
Calculatepercentage[term]=(float)totalMarks[t
subjects,max100markseach,total500).
Foreachstudent(0to19):
PrintID,Name.
Foreachterm(0to3):
PrintTerm#,TotalMarks,Percentage.
Stop.
```

CProgramtoGenerateProgressReportusingStructu

C

```
#include<stdio.h>
#include<string.h>
structStudent{
intstudentID;
charname[50];
intmarks[4][5];
inttotalMarks[4];
```

```
floatpercentage[4];
};
intmain(){
constintNUM_STUDENTS=20;
constintNUM_TERMS=4;
constintNUM_SUBJECTS=5;
structStudentclassX[NUM_STUDENTS];
printf("--EnterStudentDataandMarks--\n");
for(inti=0;i<NUM_STUDENTS;i++){
printf("\nEnterdetailsforStudent%d:\n",i+1);
printf("StudentID:");
scanf("%d",&classX[i].studentID);
getchar();
printf("Name:");
fgets(classX[i].name,sizeof(classX[i].name),stdi
classX[i].name[strcspn(classX[i].name,"\n")]=0;
```

```
for(intj=0;j<NUM_TERMS;j++){
classX[i].totalMarks[j]=0;
printf("Entermarksfor%sforTerm%d(outof100pe
subject):\n",classX[i]name,j+1);
for(intk=0;k<NUM_SUBJECTS;k++){
printf("Subject%d:",k+1);
scanf("%d",&classX[i].marks[j][k]);
classX[i].totalMarks[j]+=classX[i].marks[j][k];
}
classX[i].percentage[j]=(float)classX[i].totalMar
(NUM_SUBJECTS*100)*100;
}
}
printf("\n\n--ClassXProgressReport--\n");
for(inti=0;i<NUM_STUDENTS;i++){
printf("\n---\n");
printf("StudentID:%d\n",classx[i].studentID);
printf("Name:%s\n",classx[i].name);
for(intj=0 ;j<NUM_TERMS;j++){
printf("Term%d:\n",j+1 );
printf("TotalMarks:%d/%d\n",classx[i].totalMark
 NUM_SUBJECTS*100);
printf("percentage:%.2f%%\n",classX[i].percentage
printf("\n");
 }
 }
return0;
 }
```

Questionno03:WriteaCprogramtogeneratethefollow

```
*
* *
* * *
* * * *

* * * * *
```

Answer:

CProgramtoGeneratepattern

C

```
#include<stdioh>
intmain(){
introws=5;
for(inti=1 ;i<=rows;i++){
for(intj=1 ;j<=i;j++){
printf("* ");
}
printf("\n");
}
return0 ;
}
```

questionno4:WriteaCprogramtoperformthefollowi

=A*(B+C),whereA,BandCarematricesof(3X3)sizeand

resultantmatrix.

Answer:

CProgramforMatrixOperationsD=A*(B+C)

```
#include<stdioh>

#defineSIZE3

voidreadMatrix(intmatrix[SIZE][SIZE],charnam

printf(“EnterelementsforMatrix%c:\n“,name);

for(inti=0;i<SIZE;i++){

for(intj=0;j<SIZE;j++){

printf(“Enterelement[%d][%d]:“,i,j);

scanf(“%d“,&matrix[i][j]);

}

}

}
```

```
voidprintMatrix(intmatrix[SIZE][SIZE],Charna
printf("\nMatrix%c:\n",name);
for(inti=0 ;i<SIZE;i++){
for(intj=0 ;j<SIZE;j++){
printf("%5d",matrix[i][j]);
}
printf("\n");
}
}
voidaddMatrices(intmat1 [SIZE][SIZE],intmat2[S
for(inti=0 ;i<SIZE;i++){
for(intj=0 ;j<SIZE;j++){
result[i][j]=mat1 [i][j]+mat2[i][j];
}
}
}
```

```
voidmultiplyMatrices(intmat1 [SIZE][SIZE],intm
for(inti=0 ;i<SIZE;i++){
for(intj=0 ;j<SIZE;j++){
result[i][j]=0 ;
for(intk=0 ;k<SIZE;k++){
result[i][j]+=mat1 [i][k]* mat2[k][j];
}
}
}
}
intmain(){
intA[SIZE][SIZE],B[SIZE][SIZE],C[SIZE][SIZE]
intsumBC[SIZE][SIZE];
intD[SIZE][SIZE];
readMatrix(A,A);
readMatrix(B,B);
readMatrix(C,C);
addMatrices(B,C,sumBC);
printf("\n--IntermediateResult(B+C)---");
printMatrix(sumBC,S);
multiplyMatrices(A,sumBC,D);
printf("\n--FinalResultD=A*(B+C)---");
printMatrix(D,D);
return0;
}
```

Questionno5:UsetheconceptofFileHandling,toWritea

C,tocollectalistofNnumbersinafile,andseparatethe

fromthegivenlistofNnumbers,andputthemintwosepar

even_fileandodd_file,respectively.

Answer:

AlgorithmforSeparatingEven/OddNumbersinFiles

Start.

```
Declarefilepointers:input_file,even_file,odd_file.
Declarevariables:N(numberofelements),num(cur
Openinput.txtinwritemode("w").
PromptuserforNnumbers.
Forifrom0 toN-1 :
Readnum.
 Writenumtoinput_file.
 Closeinput_file.
 Openinput.txtinreadmode("r").
 Openeven_file.txtinwritemode("w").
Openodd_file.txtinwritemode("w").
Loopwhilefscanfsuccessfullyreadsanumberfromin
Ifnum%2==0 :writenumtoeven_file.
 Else:writenumtoodd_file.
Closeallfiles.
 Stop.
```

CProgramforSeparatingEven/OddNumbersinFiles

C

```
#include<stdioh>
#include<stdlib.h>//Forexit()
intmain(){
FILE*input_file,*even_file,*odd_file;
intN,num;
//Createinput.txtandpopulateitwithNnumbers
input_file=fopen(“input.txt“,“w“);
if(input_file==NULL){
perror(“Errorcreatinginput.txt“);
return) ;
}
printf(“Enterthenumberofintegers(N):“);
scanf(“%d“,&N);
printf(“Enter%dintegers:\n“,N);
```

```
 for(inti=0 ;i<N;i ++){
scanf("%d",&num);
 fprintf(input_file,"%d\n",num);
}
 fclose(input_file);
 printf("Numberswrittentoinput.txt\n");
//Openfilesforreadingandwritingeven/oddnumbers
input_file=fopen("input.txt","r");
 even_file=fopen("even_file.txt","w");
 odd_file=fopen("odd_file.txt","w");
if(input_file==NULL||even_file==NULL||odd_file==NU
perror("Erroropeningfiles");
return1 ;
 }
//Readfrominput_fileandseparatenumbers
 while(fscanf(input_file,"%d",&num)==1 ){
```

```
if(num%2==0){
fprintf(even_file,"%d\n",num);
}else{
fprintf(odd_file,"%d\n",num);
}
}
printf("Numbersseparatedintoeven_filetxtandodd
//closeallfiles
fclose(input_file);
fclose(even_file);
fclose(odd_file);
return0;
}
```

Questionno6:WritePythoncodetoperformthefollowi
firsttxttosecondtxt(ii)Readingafile(iii)Writingin
intoafile

Answer:

PythonCodeforFileOperations

Python

```
#(i)Copycontentoffilefirst.txttosecond.txt
try:
withopen("first.txt","w")asf_first:
f_first.write("Thisisthecontentoffirst.txt\nLine2
withopen("first.txt","r")asf_in:
content=f_in.read()
withopen("second.txt","w")asf_out:
f_out.write(content)
print("Contentcopiedfromfirst.txttosecond.txt")
exceptFileNotFoundError:
print("Error:first.txtnotfoundforcopying")
#(ii)Readingafile(eg,second.txt)
try:
```

```
withopen("second.txt","r")asf_read:
print("\n--Readingsecond.txt--")
print(f_read.read())
exceptFileNotFoundError:
print("Error:second.txtnotfoundforreading")
#(iii)Writingintoafile(overwritesexistingcontent
withopen("output.txt","w")asf_write:
f_write.write("Thisisnewcontentwrittentooutput.tx
f_write.write("Thislineoverwritespreviouscontent\n
print("\nNewcontentwrittentooutput.txt")
#(iv)Appendingintoafile
withopen("output.txt","a")asf_append:
f_append.write("Thislineisappendedtooutput.txt\n")
f_append.write("Anotherappendedline\n")
print("Contentappendedtooutput.txt")
#Verifyappendedcontentbyreadingoutput.txt
```

```
try:
withopen("output.txt","r")asf_verify:
print("\n--Verifyingoutput.txtafterappend--")
print(f_verify.read())
exceptFileNotFoundError:
print("Error:output.txtnotfoundforverification
```

questionno.7:Writeanalgorithmtofindtheslopeofaline

endpointcoordinatesare(x1 ,y1)and(x2 ,y2).Thealgo

slopeispositive,negativeorzero.Transformyouralgo

Answer:

```
AlgorithmtoFindSlopeandItsSign
Start.
Inputx1 ,y1 ,x2 ,y2.
Checkforverticalline:Ifx2 x1 ==0 :
 Ify2 y1 ==0 :Print"Slopeisundefined(pointsareide
Else:Print"Slopeisundefined(verticalline)".
```

Gotostep7.

Calculateslope:slope=(y2-y1))/(x2-x1)).

Determineslopesign:

Ifslope>0:Print"Slopeispositive".

Elseifslope<0:Print"Slopeisnegative".

Else(slope==0):Print"Slopeiszero".

Print"CalculatedSlope:",slope.

Stop.

PythonProgramforSlopeCalculation

Python

```
defcalculate_slope_and_sign(x1,y1,x2,y2):
"""
Calculatestheslopeofalinesegmentanddeterminesifi
negative,orzero.
Handlesverticallines.
"""
```

```
delta_x=x2 x1
delta_y=y2 y1
ifdelta_x==0:
ifdelta_y==0:
print("Slopeisundefined(pointsareidentical)")
else:
print("Slopeisundefined(verticalline)")
returnNone#ReturnNoneforundefinedslope
else:
slope=delta_y/delta_x
print(f"CalculatedSlope:{slope}")
ifslope>0:
print("Slopeispositive")
elifslope<0:
print("Slopeisnegative")
else:
```

```
print("Slopeiszero")
returnslope
#Tryouts
print("---TestCase1:PositiveSlope---")
calculate_slope_and_sign(1,2,3,4)#Slope=(4-2)
print("\n---TestCase2:NegativeSlope---")
calculate_slope_and_sign(1,4,3,2)#Slope=(2-4
print("\n---TestCase3:ZeroSlope(HorizontalLine)-
calculate_slope_and_sign(1,2,5,2)#Slope=(2-2)
print("\n---TestCase4:UndefinedSlope(VerticalLin
calculate_slope_and_sign(2,1,2,5)#Slope=(5-1)/
print("\n---TestCase5:UndefinedSlope(IdenticalP
calculate_slope_and_sign(2,2,2,2)#Slope=(2-2)/
```

questionno8:WriteaprogrammeinPythontocreatepa

3moduleinitnamedCube,CuboidandSphereeachhavinga

VolumeofCube,CuboidandSphererespectivelyImportth locationandusethefunctionsAssumptionscanbemadew

yourprogrammewithsuitablecommentstoimproverea

Answer:

PythonProgramforVolumepackage

DirectoryStructure:

my_project/

main_program.py

Volume/

init.py

Cube.py

Cuboid.py

Sphere.py

Contentofvolume/_init_.py(Emptyorcanlistmodules

Python

#ThisfilemakesVolumeaPythonpackage.

#Itcanbeempty,oryoucanimportspecificfunctions

#fromitssub-modulestomakethemdirectlyaccessibl

#Forthisexample,wellkeepitsimpleandjustmakeitap

Contentofvolume/cube.py:

Python

```
#volume/cube.py
defcalculate_cube_volume(side):
"""
Calculatesthevolumeofacube.
Args:
side(floatorint):Thelengthofonesideofthecube.
Returns:
float:Thevolumeofthecube.
"""
returnside**3
if__name__=='__main__':
#Exampleusagewhenrunningthismoduledirectly
```

```
print(f"Volumeofcubewithside5:{calculate_cube_v
Contentofvolume/cuboid.py:
Python
#volume/cuboid.py
defcalculate_cuboid_volume(length,width,height):
"""
Calculatesthevolumeofacuboid.
Args:
length(floatorint):Thelengthofthecuboid.
width(floatorint):Thewidthofthecuboid.
height(floatorint):Theheightofthecuboid.
Returns:
float:Thevolumeofthecuboid.
"""
returnlength*width*height
if_name_=='_main_:'
#Exampleusagewhenrunningthismoduledirectly
```

print(f"Volumeofcuboid2x3x4:{calculate_cuboid_

ContentofVolume/Spherep.y:

Python

```
#Volume/Spherep.y

importmath

defcalculate_sphere_volume(radius):

""""""

Calculatesthevolumeofasphere.

Args:

radius(floatorint):Theradiusofthesphere.

Returns:

float:Thevolumeofthesphere.

""""""

return(4/3)*mathp.i*(radius**3)

if_name_=='_main_':

#Exampleusagewhenrunningthismoduledirectly
```

```
print(f"Volumeofspherewithradius3:{calculate_sph
```

Contentofmain_programpy(separatelocation,importi

Python

```
#main_programpy
 #Thisscriptdemonstrateshowtoimportandusefunc
#option1:Importspecificfunctionsfrommodules
 fromVolumeCubeimportcalculate_cube_volume
fromVolumeCuboidimportcalculate_cuboid_volume
fromVolumeSphereimportcalculate_sphere_volume
 print("—Calculatingvolumesusingimportedfunctio
 #Usetheimportedfunctions
cube_vol=calculate_cube_volume(7)
 print(f"VolumeofCube(side7):{cube_vol}")
cuboid_vol=calculate_cuboid_volume(4,5,6)
print(f"VolumeofCuboid(4x5x6):{cuboid_vol}")
 sphere_vol=calculate_sphere_volume(3.5)
```

```
print(f"Volumeof Sphere(radius3.5):{sphere_vol:.2f
#Option2:Importtheentiremodule(lesscommonfor
#importVolume.Cube
#importVolume.Cuboid
#importVolume.Sphere
#print("\n—Calculatingvolumesusingmoduleobje
#cube_vol_alt=Volume.Cube.calculate_cube_volume(
#print(f"VolumeofCube(side2):{cube_vol_alt}")
#cuboid_vol_alt=Volume.Cuboid.calculate_cuboid_vol
#print(f"VolumeofCuboid() x2x3):{cuboid_vol_al
#sphere_vol_alt=Volume.Sphere.calculate_sphere_volu
#print(f"VolumeofSphere(radius) ):{sphere_vol_al
```

Questionno9:WriteaprograminPythontoperformfoll

Tofindsquarerootofnumbersinalistusinglambdafunc

Todisplayfirstnlinesfromafile,whereenisgivenbyuse

Todisplaysizeofafileinbytes

Todisplayfrequencyofeachwordinafile.

Answer:

PythonProgramforVariousOperations

Python

```
importmath
importos#Forfilesize
#--1Findsquarerootofnumbersinalistusinglambda
numbers=[1,4,9,16,25,36,49,64,81,100]
#Usemapwithalambdafunctiontoapplysqrttoeache
square_roots=list(map(lambdax:math.sqrt(x),numb
print(f"Originalnumbers:{numbers}")
print(f"Squareroots(usinglambda):{square_roots
#--2Displayfirstnlinesfromafile--
#Createadummyfilefordemonstration
withopen("sample_lines.txt","w")asf:
f.write("Line1:Helloworld\n")
```

```
f.write(“Line2:Pythonprogramming\n“)
f.write(“Line3:Filehandlingexample\n“)
f.write(“Line4:Morelinesfollow\n“)
f.write(“Line5:Endofsamplefile\n“)
file_name_lines=“sample_lines.txt“
try:
n_lines=int(input(“\nEnternumberoflinestodisplay
“))
withopen(file_name_lines,“r“)asf:
fori,lineinenumerate(f):
ifi<n_lines:
print(line.strip())#strip()removesnewlinecharacter
else:
break
exceptFileNotFoundError:
print(f“Error:{file_name_lines}notfound.“)
```

```
exceptValueError:
print("Invalidinputfornumberoflines.Pleaseenterar
#—3Displaysizeofafileinbytes—
file_name_size="sample_lines.txt"#Usingtheprevio
try:
file_size=os.pathgetsize(file_name_size)
print(f"\nSizeof{file_name_size}{file_size}bytes")
exceptFileNotFoundError:
print(f"Error:{file_name_size}notfoundtochecksiz
#—4Displayfrequencyofeachwordinafile—
#Createanotherdummyfileforwordfrequency
withopen("word_freq_sample.txt","w")asf:
f.write("Pythonisagreatlanguage.Pythonprogrammi
f.write("LearningPythonisrewarding.IsPythoneasy?"
file_name_freq="word_freq_sample.txt"
word_frequency={}
```

```
try:
withopen(file_name_freq,"r")asf:
text=f.read().lower()#Readcontentandconverttolow
#Removepunctuationandsplitintowords
words=text.replace(').replace(,').replace(?').split()
forwordinwords:
word_frequency[word]=word_frequency.get(word,
print(f"\nWordfrequencyin{file_name_freq}")
forword,countinword_frequency.items():
print(f"{word}{count}")
exceptFileNotFoundError:
print(f"Error:{file_name_freq}notfoundforword
```

Questionno.0 :WhatareCoroutines?HowCo-routinesd

routinessupportcooperativemulti-taskinginpython?Co

routines.

Answer:

Co-routines,Threads,andMulti-tasking

Coroutines:Coroutinesarefunctionsthatcanbepau
cooperativemultitasking,meaningtheexecutionofaf
certainpoints(usingyieldorawaitinPython)andlate
off.Theymanagetheirownyieldingofcontrol,typica

Coroutinesvs.Threads:

Whencontrastingcoroutinesandthreads,thefundamer
intheirapproachtoconcurrencyandhowtheyachieveit
traditional,operating-system-levelformofconcur
independentexecutionunitmanagedbytheOSscheduler
switchbetweenthreads(pre-emptivemultitasking).This
coreprocessors,butcomeswithoverhead:creatingan
expensiveintermsofmemoryandCPUcycles,andtheyint
ofraceconditionsandlocksforshareddata,asmultip
simultaneously.Incontrast,coroutinesofferalight
concurrency.Theyarefunctionswhoseexecutioncanbep

specificpoints,allowingforcooperativemultitaski

"cooperative":coroutinesexplicitlyyieldcontrolb

whentheyencounterablockingoperation(likeI/O),allo

Thismeansonlyonecoroutinerunsatatimewithinasin

needforcomplexlocksforshareddata(unlessexplic

withincoroutines).Coroutinesaresignificantlycheap

thanthreads,makingthemidealforI/O-boundtaskswh

theirtimewaiting,buttheydontprovidetrueparallel

HowCo-routinesSupportCooperativeMulti-taskinginP

(implementedusingasyncdefandawait,orolderyield

multitaskingbyexplicitlycedingcontrol.Whenaco-rou

yieldexpression,itpausesitsexecutionandreturnscon

eventloopcanthenpickanotherco-routinethatisready

multiple"tasks"tomakeprogressconcurrentlywith

overheadofcontextswitchingbetweenOSthreads.This

I/O-boundoperations,wheretasksspendmostoftheirti

Subroutinesvscoroutines

Whendistinguishingbetweencoroutinesandsubroutine
theircontrolflowandstatemanagementcapabilitie
typicalfunctionormethodinmostprogramminglangu
fromitsentrypointtoitsexitpointItfollowsaLIFO(L
executionmodel,alwaysreturningcontroltoitscall
itslocalstateisgenerallydestroyeduponcompletion
entrypointandasingleexitpointConversely,acoroutin
ofasubroutinethatallowsformultipleentryandexit
pausedatanypointwithinitsexecution(usingconstruc
returningcontroltoitscaller,butcrucially,itret
execution(context)whenityieldsWhenresumed,itconti
preciselywhereitleftoff,ratherthanstartingfr
resume"capabilitymakescoroutinessuitableforco
tomanagecomplex,non-blockingoperations,whereass
andsequential