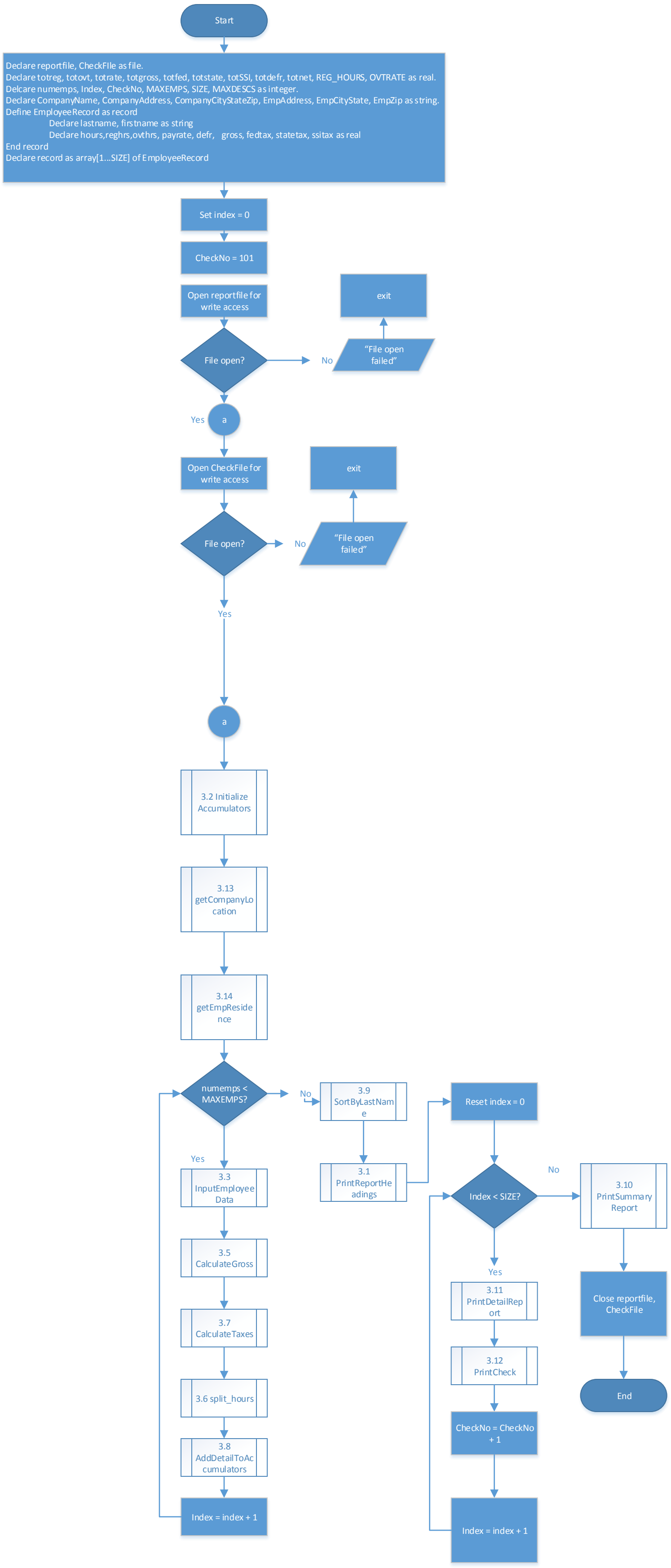
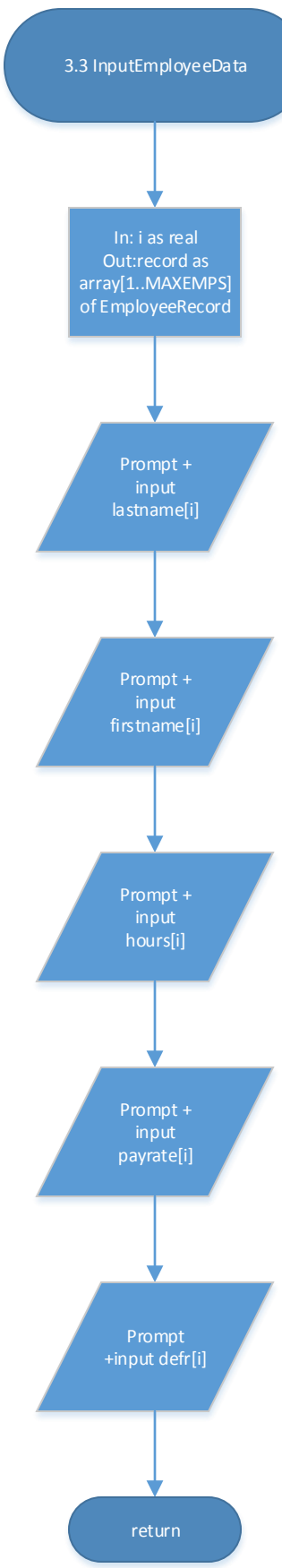
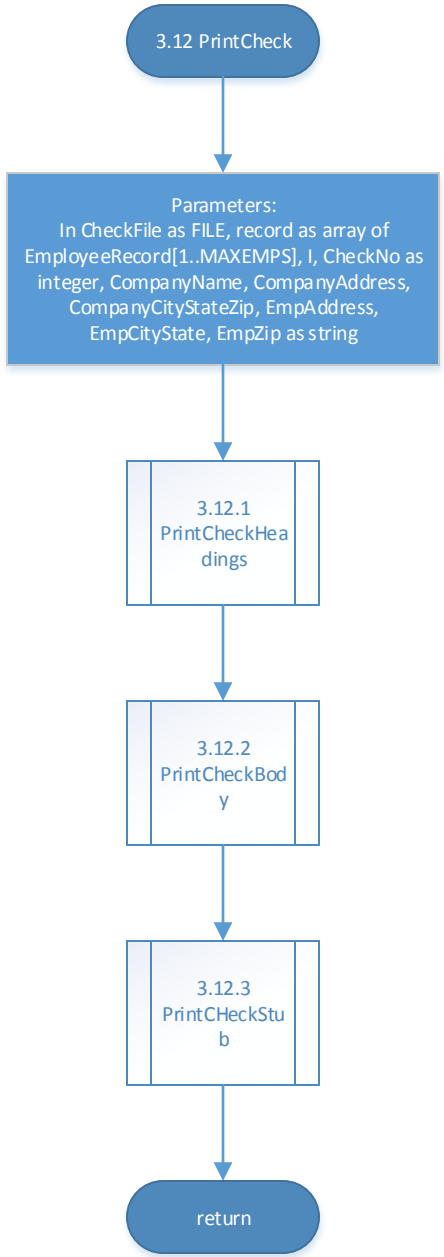
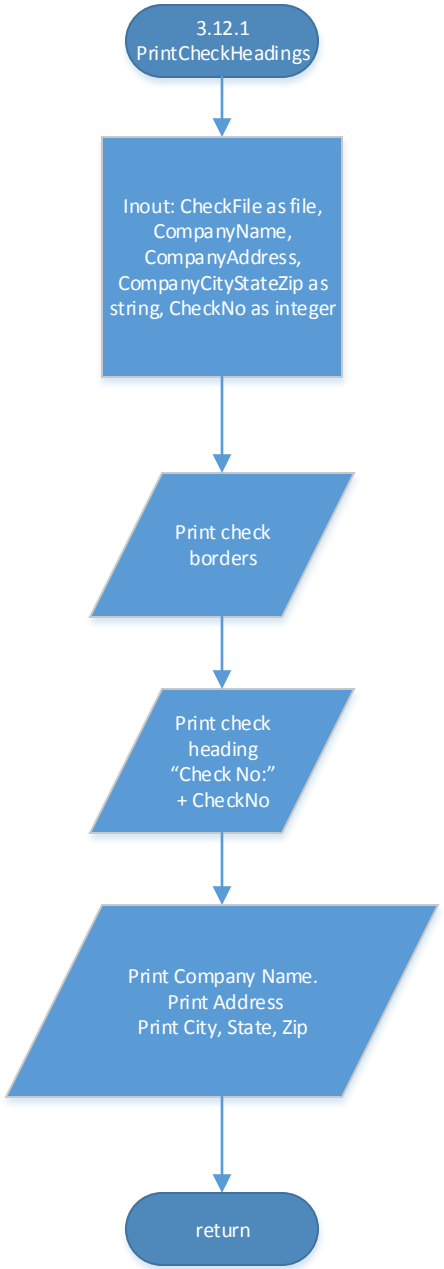
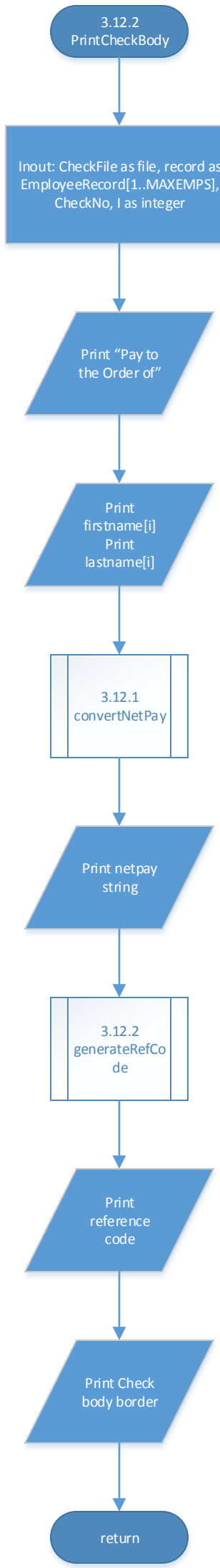
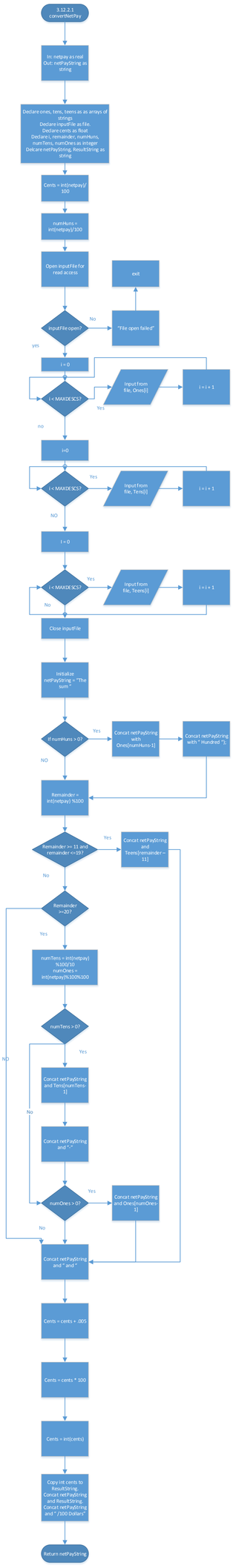

Start
Declare reportfile, CheckFIle as file.
Declare totreg, totovt, totrate, totgross, totfed, totstate, totSSI, totdefr, totnet, REG_HOURS, OVTRATE as real.
Delcare numemps, Index, CheckNo, MAXEMPS, SIZE, MAXDESCS as integer.
Declare CompanyName, CompanyAddress, CompanyCityStateZip, EmpAddress, EmpCityState, EmpZip as string.
Define EmployeeRecord as record
Declare lastname, firstname as string
Declare hours,reghrs,ovthrs, payrate, defr, gross, fedtax, statetax, ssitax as real
End record
Declare record as array[1...SIZE] of EmployeeRecord
Set index = 0
CheckNo = 101
Open reportfile for write access
File open?
No
"File open failed"
exit
Yes
a
Open CheckFile for write access
File open?
No
"File open failed"
exit
Yes
a
3.2 Initialize Accumulators
3.13 getCompanyLocation
3.14 getEmpResidence
numemps < MAXEMPS?
No
3.9 SortByLastName
3.1 PrintReportHeadings
Reset index = 0
Index < SIZE?
No
3.10 PrintSummary Report
Close reportfile, CheckFile
End
Yes
3.3 InputEmployeeData
3.5 CalculateGross
3.7 CalculateTaxes
3.6 split_hours
3.8 AddDetailToAccumulators
Index = index + 1
Yes
3.11 PrintDetailReport
3.12 PrintCheck
CheckNo = CheckNo + 1
Index = index + 1

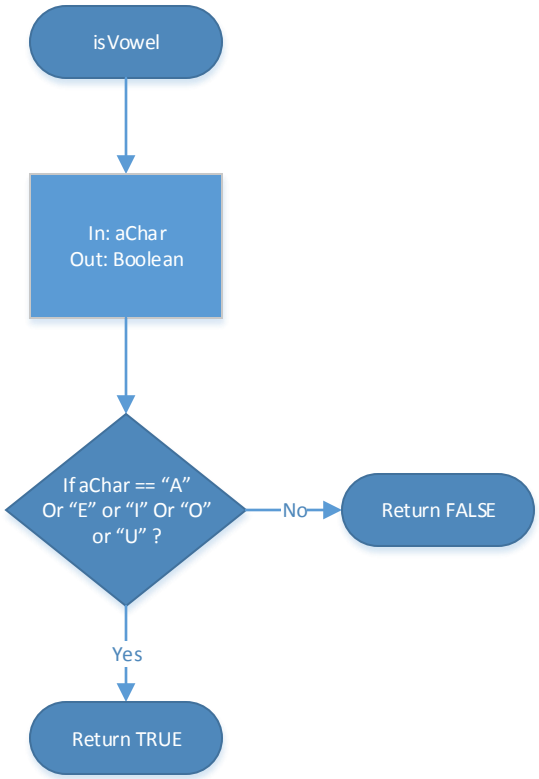

3.3 InputEmployeeData
In: i as real
Out:record as array[1..MAXEMPS] of EmployeeRecord
Prompt + input lastname[i]
Prompt + input firstname[i]
Prompt + input hours[i]
Prompt + input payrate[i]
Prompt +input defr[i]
return

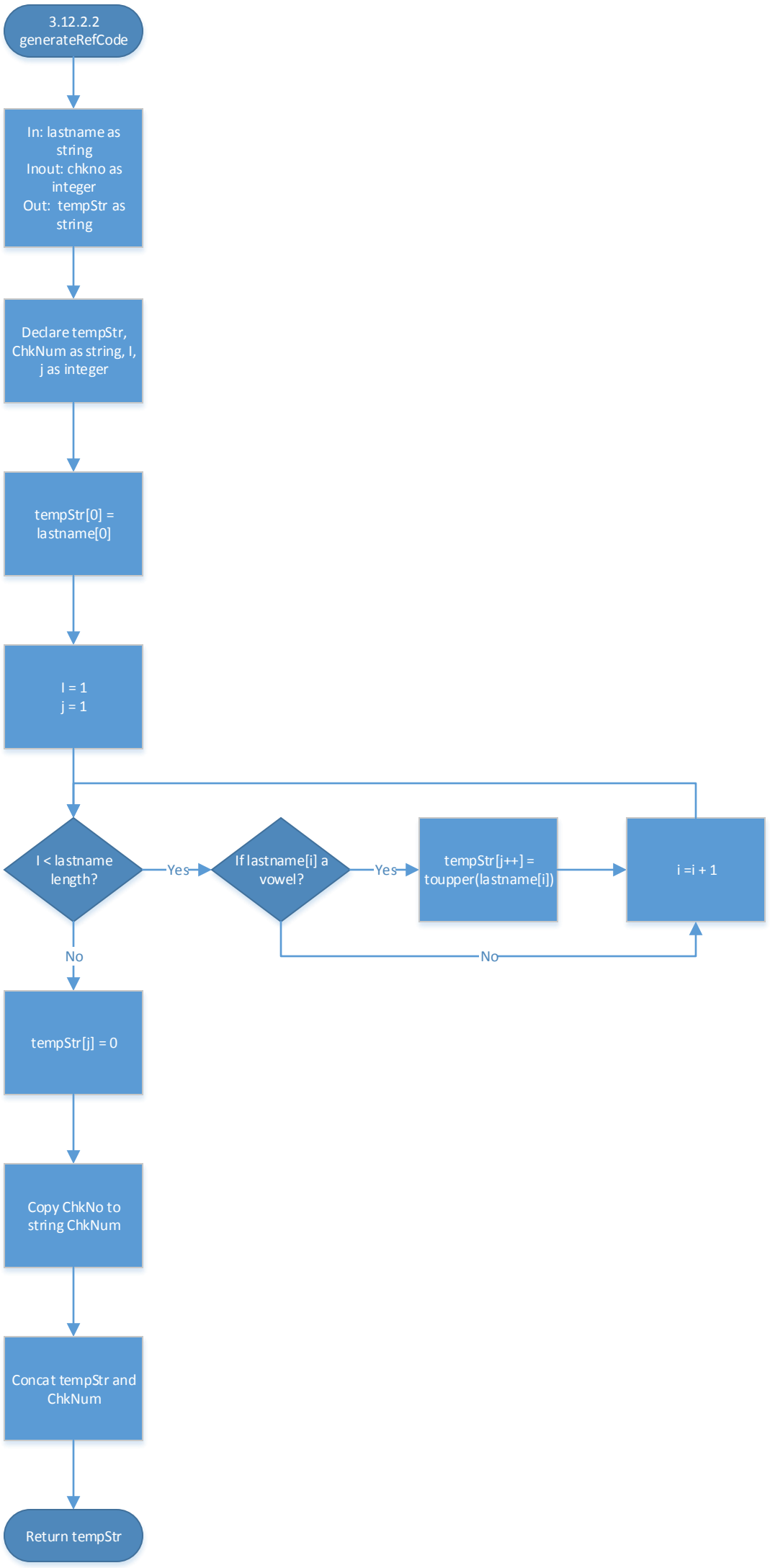

3.12 PrintCheck
Parameters:
In CheckFile as FILE, record as array of EmployeeRecord[1..MAXEMPS], I, CheckNo as integer, CompanyName, CompanyAddress, CompanyCityStateZip, EmpAddress, EmpCityState, EmpZip as string
3.12.1 PrintCheckHeadings
3.12.2 PrintCheckBody
3.12.3 PrintCHeckStub
return

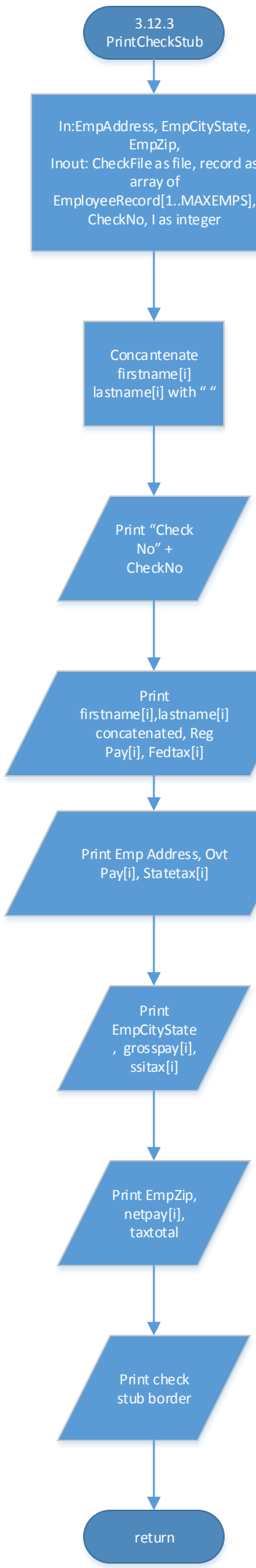

3.12.1
PrintCheckHeadings
Inout: CheckFile as file, CompanyName, CompanyAddress, CompanyCityStateZip as string, CheckNo as integer
Print check borders
Print check heading "Check No:" + CheckNo
Print Company Name.
Print Address
Print City, State, Zip
return

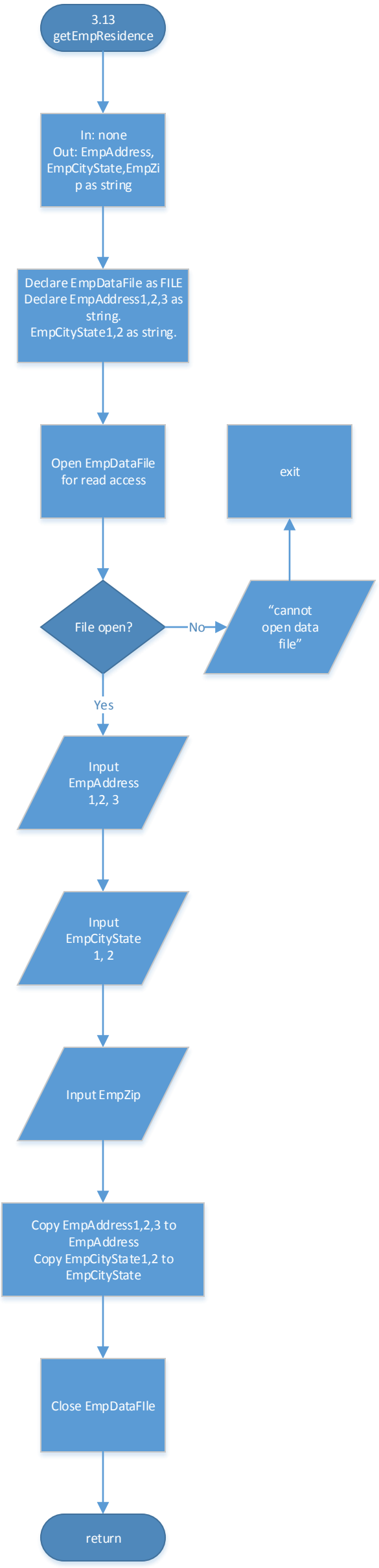

3.12.2
PrintCheckBody
Inout: CheckFile as file, record as EmployeeRecord[1..MAXEMPS], CheckNo, I as integer
Print “Pay to the Order of”
Print firstname[i]
Print lastname[i]
3.12.1
convertNetPay
Print netpay string
3.12.2
generateRefCode
Print reference code
Print Check body border
return

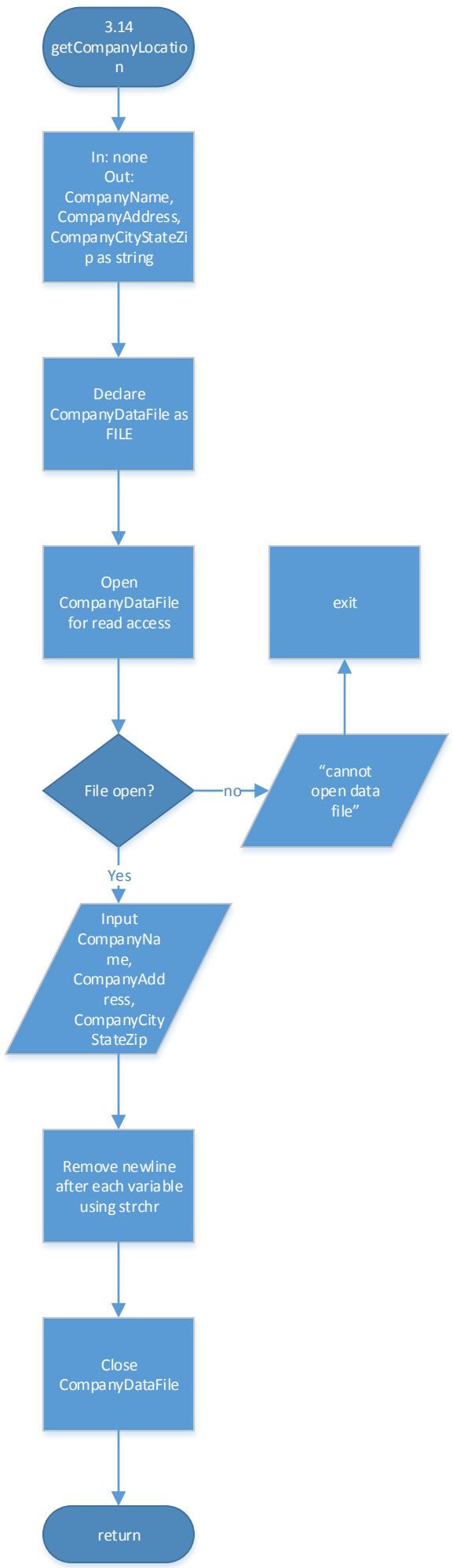

3.12.2.1 convertNetPay
In: netpay as real
Out: netPayString as string
Declare ones, tens, teens as as arrays of strings
Declare inputFile as file.
Declare cents as float
Declare i, remainder, numHuns, numTens, numOnes as integer
Delcare netPayString, ResultString as string
Cents = int(netpay)/ 100
numHuns = int(netpay)/ 100
Open inputFile for read access
exit
inputFile open?
No
"File open failed"
yes
i = 0
i < MAXDESCS?
Input from file, Ones[i]
Yes
i = i + 1
no
i=0
i < MAXDESCS?
Yes
Input from file, Tens[i]
i = i + 1
NO
I = 0
i < MAXDESCS?
Yes
Input from file, Teens[i]
i = i + 1
No
Close inputFile
Initialize netPayString = "The sum "
If numHuns > 0?
Yes
Concat netPayString with Ones[numHuns-1]
Concat netPayString with " Hundred ");
NO
Remainder = int(netpay) %100
Remainder >= 11 and remainder <=19?
Yes
Concat netPayString and Teens[remainder – 11]
No
Remainder >=20?
Yes
numTens = int(netpay) %100/10
numOnes = int(netpay)%100%100
numTens > 0?
Yes
NO
Concat netPayString and Tens[numTens-1]
No
Concat netPayString and "-"
numOnes > 0?
Yes
Concat netPayString and Ones[numOnes-1]
No
Concat netPayString and " and "
Cents = cents + .005
Cents = cents * 100
Cents = int(cents)
Copy int cents to ResultString.
Concat netPayString and ResultString.
Concat netPayString and " / 100 Dollars"
Return netPayString


isVowel
In: aChar
Out: Boolean
If aChar == “A”
Or “E” or “I” Or “O”
or “U” ?
No
Return FALSE
Yes
Return TRUE


3.12.2.2
generateRefCode
In: lastname as string
Inout: chkno as integer
Out: tempStr as string
Declare tempStr, ChkNum as string, I, j as integer
tempStr[0] = lastname[0]
I = 1
j = 1
I < lastname length?
Yes
If lastname[i] a vowel?
Yes
tempStr[j++] = toupper(lastname[i])
i =i + 1
No
No
tempStr[j] = 0
Copy ChkNo to string ChkNum
Concat tempStr and ChkNum
Return tempStr


3.12.3
PrintCheckStub
In:EmpAddress, EmpCityState, EmpZip,
Inout: CheckFile as file, record as array of EmployeeRecord[1..MAXEMPS], CheckNo, I as integer
Concantenate firstname[i] lastname[i] with " "
Print "Check No" + CheckNo
Print firstname[i],lastname[i] concatenated, Reg Pay[i], Fedtax[i]
Print Emp Address, Ovt Pay[i], Statetax[i]
Print EmpCityState , grosspay[i], ssitax[i]
Print EmpZip, netpay[i], taxtotal
Print check stub border
return


3.13
getEmpResidence
In: none
Out: EmpAddress,
EmpCityState,EmpZi
p as string
Declare EmpDataFile as FILE
Declare EmpAddress1,2,3 as
string.
EmpCityState1,2 as string.
Open EmpDataFile
for read access
exit
File open?
No
“cannot
open data
file”
Yes
Input
EmpAddress
1,2, 3
Input
EmpCityState
1, 2
Input EmpZip
Copy EmpAddress1,2,3 to
EmpAddress
Copy EmpCityState1,2 to
EmpCityState
Close EmpDataFIle
return


3.14 getCompanyLocation
In: none Out: CompanyName, CompanyAddress, CompanyCityStateZip as string
Declare CompanyDataFile as FILE
Open CompanyDataFile for read access
exit
File open?
no
“cannot open data file”
Yes
Input CompanyName, CompanyAddress, CompanyCityStateZip
Remove newline after each variable using strchr
Close CompanyDataFile
return