Onboarding Users Script

Goal: Create an LDIF entry containing required fields
Username, Firstname, Lastname, title, groups, password

Notes

- `IFS` is a shell variable that determines how strings are split, by default is space, tab, newline
- `read` is a command that parses inputs by the IFS, and stops when it reaches a newline. It then skips the newline character and proceeds to the next line if there are more lines
  - -r to prevent backslash escapes from being interpreted
  - -a to store read items into an array
- `<<\[f]` creates a heredoc, where f can be any delimiter phrase, which allows for multiple lines of input. At the end of the input, use the delimiter
- Attributes
  - `dn` (Distinguished Name): Uniquely identifies LDAP entry, explains where in the directory tree the entry resides
  - `objectClass` defines the allowed and required attributes
    - `inetOrgPerson` - Encompasses person and `organizationalPerson`, requires a `cn` attribute
    - `posixAccount` - Defines `uidNumber`, `gidNumber`, `homeDirectory`, `loginShell` for Unix logins
  - `uid` (User Identifier): Defines a unique login name associated with the user
  - `cn` (Common Name): The name of an object, and typically the full name of a person
  - `givenName` : Represents the first name of the user
  - `sn` (Surname): Represents the last name of the user
  - `title` : Stores job title of the use
  - `userPassword` : Hold's the user password. Should be stored as hash value
- a `<<<` b: Known as the here-string operator, feeds the contents of the value of b into the command a

```
#!/bin/bash

# run OnboardUsers script using ./script.sh < [csv_filename]

#set variables as needed that describe components in LDAP directory
dn="team17,dc=oak,dc=swccdc,dc=com" # [SUBJECT TO CHANGE]
userOU="ou=users,${dn}"
groupOU="ou=groups,${dn}"

#create functions to dynamically assign user IDs if needed
# [FUNCTIONS GO HERE]

#parse through individual users and create an LDIF document for them
IFS=","
while read -r username firstName lastName jobTitle groups password; do # [FIELDS SUBJECT TO CHANGE]
# store the following information into the file user.ldif
    cat > ${username}.ldif <<EOF
dn: uid=${username},${userOU}
objectClass: inetOrgPerson
objectClass: posixAccount
uid: ${username}
givenName: ${firstName}
sn: ${lastName}
cn: ${firstName} ${lastName}
```

```
title: ${jobTitle}
userPassword: ${password}
EOF

#add user to LDAP directory
    ldapadd -x -D [ADMIN_DN] -w [ADMIN_PW] -f "${username}.ldif"
    echo Account created for ${username}

#add user to correct groups
    read -ra groupArray <<< "${groups}"
    for group in "${groupArray[@]}";do # expands to all elements in array
        cat > group.ldif <<EOF
dn: cn=${group},${groupOU}
changetype: modify
add: member
member: uid=${username},${userOU}
EOF

        ldapmodify -x -D [ADMIN_DN] -w [ADMIN_PW] -f group.ldif
        echo "Added ${username} to ${group}"
    done
done
```