前提:

README.mdに示した使用ソフトのインストールおよび初期設定は完了しているものとする。

問題1

Gitはファイルやプログラムの(ア)を行うためのシステムである。Gitなどの(ア)システムを導入することによって、(イ|2つ挙げよ)ができるようになる。

(ア)システムは(ウ)システムと(エ)システムの2パターンに分けられるがGitは(エ)である。

(ウ)システムでは、複数人でファイルなどを共同編集する場合、中央サーバに直接変更後のファイルを送るが、(エ)システムでは(オ)。こうすることで、共同編集時に生じる混乱を防いでいる。また(カ)というメリットもある。

問題2

Gitの特徴としては

・ネットワークに接続していなくても作業ができる(一つ目)

・履歴を保存する前にStagingを行う(二つ目)

・他のユーザと柔軟なやり取りができる(三つ目)

が挙げられる。

一つ目･･･問題1で述べた通り、分散型は―ジョン管理システムの利点である。

二つ目･･･Gitでは、(ア)をCommitという。Commitを行う際の(イ)をStagingという。  
Stagingを行うことによって、(ウ)を一つのCommitにまとめたり、逆に単一のファイルに施した変更を(エ)たりなど、Commitを柔軟に加工し、(オ)ことが可能となる。  
(オ)が可能になることによって、(カ|バグ対策に役立つ点を2つ挙げよ)などのメリットが生まれる。

三つ目･･･Gitを用いたツール「GitHub」はみんなで使うSNSであるため、他人のソースを自由に見ることができる。また、そのソースに問題が見つかったなどの場合、修正を加えてあげて、それを反映するようにリクエストを送ることができる。これを(キ)という。

問題3  
*参考: https://wa3.i-3-i.info/word15664.html*

*リポジトリ (repository)とは―「分かりそう」で「分からない」でも「分かった」気になれるIT用語辞典 (2018年11月24日閲覧)*

リポジトリとは何か。簡単にいえ。

問題4

Gitはファイルを共同編集する際にも変更点を適切に管理でき、大変便利であるが、それでも非常の多くの人が参加するプロジェクトで、かつ(ア)場合、リポジトリの管理が複雑になる。そこで、Gitにおいて(ア)のを簡単にできるようにするためのHub(中継地点)として生れたのがGitHubである。

問題5

GitHubにおいて、PushとPullの意味をそれぞれ説明せよ。

問題6

ForkとPull RequestはGitにないGitHubの特徴的な機能の例である。ForkとPull Requestの意味に触れながら、さらにFork、Pull Request、Mergeによって「プロジェクトに参加できる」ことを説明する文章を書け。

問題7

Cloneとは何か

問題8

コマンドプロンプトを立ち上げた。

GitHubユーザ17ec084のリポジトリtestを一つ下のディレクトリcloneHereにCloneせよ。

(gitコマンドを使うためのパスは通っていて、windowsまたはLinuxの環境とする)

問題9

ローカルディレクトリaはローカルリポジトリである。a/b.javaとa/c.cを書き換えた。この変更の内、b.javaに施した変更のみを共有リポジトリAに反映させたい。コマンドプロンプトから行いたい。コマンドプロンプトを起動するところから方法を説明せよ。

問題10

現在のディレクトリ内の変更すべてを共有リポジトリに反映させたいとき、どのようなコマンドによってStagingできるかいえ

問題11

問題9に出てきたmasterについて説明せよ。

問題12

ブランチ(Branch)とは何か。

問題13

.gitignoreファイルは(ア)を指定するためのものである。

(ア)のパターンは1行ずつ記述される。

(イ)･･･(イ)から始まる行はコメントとして使用可能。

hoge･･･(ウ)

\*.c･･･(エ)

!a.c･･･但し、(オ)

/\*.java･･･(カ)

問題14

newfile.txtはすでにGitで管理されている。このような場合、後から .gitignoreで無視するように指定しても引き続き管理される。それでも無視させなければいけない場合、どうするか。

問題15

(1)Gitの設定をいじるコマンドを答えよ。

また、そのコマンドを用いて改行コードの設定(Commit時にCRLF→LF(またはその逆))を

(2)デフォルトで行え

(3)特定のリポジトリのみで行え

問題16

答え1

(ア):バージョン管理

(イ):変更の履歴の確認と、過去の特定の状態へ戻すこと

(ウ):中央集権型バージョン管理

(エ):分散型バージョン管理

(オ):全員がそれぞれローカル環境に「中央サーバのコピー」を作り、その差分を本物の中央サーバに送る

(カ):ネットワークに接続していなくても作業ができる

答え2

(ア):変更履歴の単位

(イ):準備

(ウ):複数のファイルに施した変更

(エ):複数のCommitに分け

(オ):各Commintに意味を持たせる

(カ):

・バグが発見されたときの切り戻しが容易になる

・どの時点でバグが入ったのかが確かめやすくなる

(キ):Pull Request

答え3

データの保管場所

答え4

(ア):複数人でリポジトリの管理を行う

答え5

Push･･･ローカルで加えた変更を共有リポジトリに反映

Pull･･･最新の共有リポジトリをローカルに反映

答え6

興味のある(他人の)プロジェクトをForkすることで、そのプロジェクトのコピーを自分のリポジトリに作る。そして加筆・修正したものを、元のプロジェクトの開発者にPull Requestとして送る。開発者は送られたPull Requestに問題がないと判断すればその変更を反映(Merge)する。この方法で、開発者でなくても興味あるプロジェクトに参加することができる。

補足:ここでいう「開発者」というのは、プロジェクトを編集する権利(コミット権)が与えられている者のことをいう。

答え7

共有リポジトリを各開発者のローカル環境にコピーすること。

答え8

cd cloneHere

git clone git@github.com:17ec084/test.git

答え9

1. ディレクトリaでコマンドプロンプトを起動する。

2. git remote -vを実行し、共有リポジトリの「名前」を確認する。  
今回はoriginであったと仮定。

2. git add b.java

3. git commit -m (コミットメッセージ)

4. git push origin master

git pushコマンドについて紹介したので、git pullコマンドの紹介は省く。引数は同じ。

答え10

git add .

答え11

masterは、レポジトリにデフォルトで与えられるブランチの名前である。

補足:とりあえず、「ブランチ」をいじっていないのであれば今のブランチは絶対masterブランチなので、言われた通りにmasterと打てばよい。

答え12

開発中のリポジトリの本流から分岐し、本流に影響を与えない形で独自の変更を加えるための「枝分かれ」における「枝」である。

答え13

(ア):Gitの管理対象外のファイル

補足: .gitignoreで指定されたファイルが書き換えられたリ消されたり作られたりしても、そのことが共有されることはない。

(イ):#

(ウ):hogeという名前のファイルを指定する

(エ): .cで終わる名前のファイルを指定する

(オ): a.cは指定から除外する

(カ): .gitignoreファイルと同じディレクトリにあるファイルで、かつ .javaで終わる名前のファイルの指定する

答え14

①.gitignoreでnewfile.txtを指定

②commit

③newfile.txtを削除する

④commit

答え15

(1)git config

(2)git config --global core.autcrlf true

(3)git config --local core.autcrlf true

または

　git config core.autcrlf true