

株式会社ナレッジフロー 社員採用向け 入社試験（高校生向け）

問題1

以下のサイトは、文章を投稿する機能を備え、投稿された文章の最新の20件を表示するWebサイトです。

<https://exam.1-mise.net/test1/test1>

C#で簡単な画面を作り、その画面で入力した文章を、上記サイトに投稿するプログラムを作成してください。

【注意事項】

通信には System.Net.Http.HttpClient を使用してください。（ブラウザコンポーネントを使用して上記サイトをブラウザコンポーネントで表示すれば同じことができますがその方法はとらないでください。）

【備考】

Webサイトへの投稿は自由に行っていただいて構いません。（問題2, 3も同様）

問題2

以下のサイトに対し、問題1と同様のことを行うプログラムを作成してください。

<https://exam.1-mise.net/test2/>

問題3

以下のサイトに対し、問題1と同様のことを行うプログラムを作成してください。

<https://exam.1-mise.net/test3/>

問題4

Eメールを受信する機能を備え、本日のEメール受信件数と本文の文字数合計を表示するWebサイトを作成してください。

参考サイト

<https://exam.1-mise.net/test1/test4>

問題5

以下のJavaScriptの2つのプログラムの結果が異なるようなAとBの具体例をあげてください。

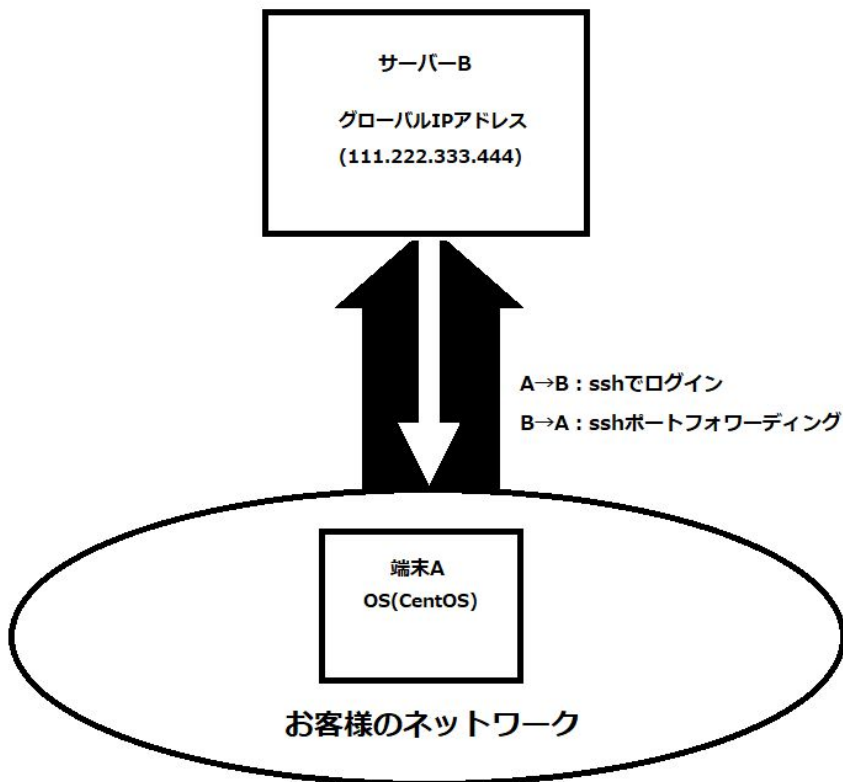
```
alert(A(function() {  
    return (function(x) {  
        return x;  
    })(B());  
}));
```

```
alert((function(x) {  
    return A(function() {  
        return x;  
    });  
})(B()));
```

問題6

CentOSがインストールされている端末Aを、お客様のネットワークに設置しているとします。

端末Aをリモートでメンテナンスするために、グローバルIPアドレス（111.222.333.444）を持つサーバーB（OSはCentOS）から端末Aにsshでログインすることを考えます。具体的には、端末AからサーバーBにsshログインし、その接続を使ってsshポートフォワーディングでサーバーBから端末Aにsshでログインすることを考えます。



端末Aが再起動したり、端末AとサーバーBでの間のネットワークに一時的な断線が起こった場合でも、再起動後もしくはネットワーク復旧後に自動的に上記のリモートメンテナンスの仕組みが復旧するようにするために、どのように設定しますか？（もしくは、どのようなソフトウェアをインストールしますか？／どのようなスクリプトを作成しますか？）具体的に記してください。

問題7

以下は、Androidで通話を自動的に録音しファイルに保存するプログラムのソースコードです。

```
public class MainActivity extends AppCompatActivity {
    private static final int PERMISSION_REQUEST_CODE = 12;
    MediaRecorder recorder = null;
    String filePath = Environment.getExternalStorageDirectory() + "/sample.mp4";

    private PhoneStateListener phoneStateListener = new PhoneStateListener() {
        @Override
        public void onCallStateChanged(int state, String phoneNumber) {
            switch (state) {
                case TelephonyManager.CALL_STATE_IDLE:
                    try {
                        if (recorder != null) {
                            releaseRecorder();
                        }
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    break;

                case TelephonyManager.CALL_STATE_OFFHOOK:
                    try {
                        if (recorder != null) {
                            releaseRecorder();
                        }
                        recorder = new MediaRecorder();
                        recorder.setAudioSource(MediaRecorder.AudioSource.DEFAULT);
                        recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
                        recorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);
                        recorder.setOutputFile(filePath);
                        recorder.prepare();
                        recorder.start();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    break;

                case TelephonyManager.CALL_STATE_RINGING:
                    break;
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        List<String> permissionList = new ArrayList<>(Arrays.asList(
```

```
        Manifest.permission.READ_PHONE_STATE,
        Manifest.permission.RECORD_AUDIO,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    ));

    String[] permissions = permissionList.toArray(new String[0]);
    if (!hasSelfPermissions(this, permissions)) {
        ActivityCompat.requestPermissions(this, permissions, PERMISSION_REQUEST_CODE);
    }

    TelephonyManager telephonyManager = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
    telephonyManager.listen(phoneStateListener, PhoneStateListener.LISTEN_CALL_STATE);
}

public boolean hasSelfPermissions(Context context, String... permissions) {
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        return true;
    }
    for (String permission : permissions) {
        if (context.checkSelfPermission(permission) != PackageManager.PERMISSION_GRANTED) {
            return false;
        }
    }
    return true;
}

public void releaseRecorder() {
    recorder.stop();
    recorder.reset();
    recorder.release();
    recorder = null;
}
}
```

このプログラムはAndroidの過去のバージョンでは正しく動作していました。

このプログラムを最近のAndroidで動かす場合にどのような問題や制約・制限などがあるか調べ、調べた結果をまとめてください。
