Odgovori na zadatke 1-3:

1Zadatak SCP

```
import paramiko


def generate_ssh_key_pair():
    # Generate an SSH key pair
    key = paramiko.RSAKey.generate(2048)

    # Save the private key to a file
    private_key_path = 'private_key.pem'
    key.write_private_key_file(private_key_path)

    # Save the public key to a file
    public_key_path = 'public_key.pub'
    with open(public_key_path, 'w') as f:
        f.write(f'ssh-rsa {key.get_base64()}')

    print(f"SSH key pair generated. Private key saved to '{private_key_path}' and public key saved to '{public_key_path}'.")


generate_ssh_key_pair()
```

Objašnjenje:

Kod koristi biblioteku paramiko, koja pruža SSH funkcionalnost u Pythonu.

Definiramo funkciju generate_ssh_key_pair() za enkapsulaciju procesa generiranja ključa.

Poziv paramiko.RSAKey.generate(2048) generiše novi RSA par ključeva sa dužinom ključa od 2048 bita.

Privatni ključ se pohranjuje u datoteku pod nazivom private_key.pem pomoću metode write_private_key_file() objekta ključa.

Javni ključ se sprema u datoteku pod nazivom public_key.pub pisanjem ključa u OpenSSH formatu, sa prefiksom ssh-rsa.

Konačno, funkcija ispisuje poruku koja potvrđuje uspješno generiranje i spremanje para ključeva.

Napomena: U ovom primjeru koristimo biblioteku paramiko radi jednostavnosti. Međutim, postoje i druge biblioteke i alati dostupni za generisanje SSH parova ključeva u različitim programskim jezicima.

C++ verzija:

```
#include <iostream>

#include <fstream>

#include <libssh/libssh.h>


void generate_ssh_key_pair() {

   // Generate an SSH key pair

   ssh_key key;

   ssh_key_new(&key);

   ssh_key_generate(key, SSH_KEYTYPE_RSA, 2048);


   // Save the private key to a file

   const char* private_key_path = "private_key.pem";

   ssh_pki_export_privkey_to_file(key, nullptr, nullptr, nullptr, private_key_path);


   // Save the public key to a file

   const char* public_key_path = "public_key.pub";

   ssh_string public_key_string = nullptr;

   ssh_pki_export_pubkey_base64(key, &public_key_string);

   const char* public_key_base64 = ssh_string_to_char(public_key_string);

   std::ofstream public_key_file(public_key_path);

   public_key_file << "ssh-rsa " << public_key_base64;

   public_key_file.close();


   std::cout << "SSH key pair generated. Private key saved to '" << private_key_path << "' and public key saved to '" << public_key_path << "'." << std::endl;


   // Clean up

   ssh_key_free(key);
```

```
    ssh_string_free_char(public_key_base64);
}

int main() {
    generate_ssh_key_pair();
    return 0;
}
```

Objašnjenje:

C++ kod koristi biblioteku libssh za generiranje SSH para ključeva i rad sa SSH funkcionalnošću.

Funkcija generate_ssh_key_pair() generiše novi RSA par ključeva dužine 2048 bita.

Privatni ključ se sprema u datoteku pomoću funkcije ssh_pki_export_privkey_to_file().

Javni ključ se izvozi u base64 kodiran niz pomoću funkcije ssh_pki_export_pubkey_base64().

Javni ključ kodiran base64 se upisuje u datoteku.

Na konzolu se ispisuje poruka koja ukazuje na uspješno generiranje i spremanje para ključeva.

Funkcija main() poziva gene_ssh_key_pair() da izvrši proces generiranja i spremanja para ključeva.

Čišćenje memorije se izvodi pomoću odgovarajućih libssh funkcija.

Napomena: Da biste kompajlirali i pokrenuli C++ kod, morate se povezati sa libssh bibliotekom i osigurati da je biblioteka instalirana na vašem sistemu. Specifičan proces kompilacije i povezivanja može varirati u zavisnosti od vašeg razvojnog okruženja i operativnog sistema.

2Zadatak SCP

```
import paramiko
import scp

def encrypt_file_with_scp(file_path, destination_host, ssh_public_key):
    # Establish an SSH connection to the destination host
    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
    ssh_client.connect(destination_host)

    # Create an SCP client
    scp_client = scp.SCPClient(ssh_client.get_transport())

    # Encrypt the file using SCP and transfer it to the destination host
    encrypted_file_path = f'{file_path}.enc'
    scp_client.put(file_path, encrypted_file_path, preserve_times=True, mode='0600')

    # Close the SCP client and SSH connection
    scp_client.close()
    ssh_client.close()

    print(f"File '{file_path}' encrypted and transferred to '{destination_host}:{encrypted_file_path}'.")

# Usage example:
file_path = 'path/to/source/file.txt'
destination_host = 'example.com'
ssh_public_key = 'path/to/public_key.pub'
encrypt_file_with_scp(file_path, destination_host, ssh_public_key)
```

Objašnjenje:

Kod koristi paramiko i scp biblioteke za SSH i SCP funkcionalnost, respektivno.

Funkcija encrypt_file_with_scp() uzima putanju datoteke, odredišni host i SSH javni ključ kao ulazne parametre.

Uspostavlja SSH vezu sa odredišnim hostom koristeći dati javni ključ.

On kreira SCP klijenta koristeći SSH transport.

Datoteka se šifrira i prenosi na odredišni host koristeći put() metodu SCP klijenta. Šifrovana datoteka je sačuvana pod istim imenom, ali sa ekstenzijom .enc.

Na kraju, funkcija ispisuje poruku koja potvrđuje uspješno šifriranje i prijenos datoteke.

3Zadatak SCP

```
import paramiko
import scp

def decrypt_file_received_via_scp(received_file_path, private_key_path):
    # Establish an SSH connection to the remote host
    ssh_client = paramiko.SSHClient()
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect('example.com')

    # Create an SCP client
    scp_client = scp.SCPClient(ssh_client.get_transport())

    # Download the encrypted file from the remote host
    local_file_path = 'decrypted_file.txt'
    scp_client.get(received_file_path, local_file_path)

    # Load the private key
    private_key = paramiko.RSAKey.from_private_key_file(private_key_path)

    # Decrypt the file using the private key
    decrypted_content = private_key.decrypt_file(local_file_path)

    # Print the decrypted content
    print(f"Decrypted content:\n{decrypted_content}")

    # Close the SCP client and SSH connection
    scp_client.close()
    ssh_client.close()
```

Usage example:

received_file_path = 'path/to/received/encrypted_file.enc'

private_key_path = 'path/to/private_key.pem'

decrypt_file_received_via_scp(received_file_path, private_key_path)

Objašnjenje:

Kod koristi paramiko i scp biblioteke za SSH i SCP funkcionalnost, respektivno.

Funkcija decrypt_file_received_via_scp() uzima putanju primljene datoteke i putanju privatnog ključa kao ulazne parametre.

Uspostavlja SSH vezu sa udaljenim hostom.

On kreira SCP klijenta koristeći SSH transport.

Šifrirana datoteka se preuzima sa udaljenog hosta pomoću metode get() SCP klijenta i sprema se kao decrypted_file.txt u lokalnom direktoriju.