자동화 만들어 보자.

1번) 이미지(10.png)는 구글 시트(Google Sheets)야. 여러 대사관에 보낼 명단이지.

2번 { }에 든 명령은 1시간마다 하루에 최대 10개씩 자동화로 보낸다는 명령어야.

2번) 명령어에서 수정해야 할 것은 2시간마다 하루에 최대 10개씩 보내는거야.

즉, 1시간에서 2시간 단위로 하루에 최대 10통만 보내는거야.

3번)은 One-Page 54 KB AI Brief.pdf 첨부파일이야. 구글 드라이브에 넣어놨어.

당연히 이메일에 첨부파일 넣고 전송해야겠지.

4번)은 스팸이 걸리지 않도록 내용을 아주 조금씩 바꿔줘. 10개의 대사관이라면 내용을 미세하게 달리해서 스팸에 걸리지 않도록 해달라는거야.

5번) 스크립트는 오늘부터 2시간 단위로 보내는거야.

아래 명령어를 분석해서 자동화 명령어를 노르웨이 대사관에 맞게 작성해주고,

테스트로 보낼 수 있게 만들어줘.

즉, 우선, 정상적으로 발송이 되는지 테스트 코드도 있어야 돼.

[ ] 아래 코드를 노르웨이 대사관용으로 바꿔줘.

노르웨이 대사관용 외교 서신은 { }야.

[ ] 안에서 아래는 이메일 제목이야.

Proposal: 54 KB Offline AI for Disability-Inclusive Protection

[ ] 안에서 아래에서부터 이메일 내용이야.

Your Excellency,

너가 할 일은

첫째, < > 이 안에 든 내용은 노르웨이 대사관 이메일 주소야. 구글 시트(Google Sheets)를 만들어줘야 돼. 그래야 자동화시킬 수 있겠지. 구글 시트 제목은 Norway

둘째, 정상적으로 발송이 되는지 테스트 코드 만들어줘야 돼.

셋째, 자동화시켜서 각국 주재 노르웨이 대사관에 1시간 단위로 보내면 돼. 하루에 최대 10건만 보낼거야. 총 69개니까 대략 7일이면 끝나겠지. 에러 코드 발생하거나, 반송 메일은 다시 메일을 보내지 않고, 포기해야 돼. 반송 메일은 다시 시도할 필요가 없다는거야. 반송 코드 있는지 우선 확인해줘. 반송 코드 있으면 추가로 작성할 필요 없어.

우선 첫째부터 시작하자.

{

Subject: Proposal for Consideration: 54 KB Offline AI for Disability-Inclusive Child Protection

Executive Summary:  
 This initiative introduces a 54 KB offline, rights-based artificial intelligence designed to protect children with disabilities in connectivity-constrained settings. Running fully on-device with no data transmission, it has been validated on low-power smartphones with minimal resource use. The approach aligns with Norway’s leadership in humanitarian protection, disability inclusion, and digital public goods, while placing no financial or operational obligations on the Embassy. Further details and supporting materials are available at the M-Corp AI introductory page ([mcorpai.org](http://mcorpai.org)).

Your Excellency,  
 Royal Norwegian Embassy,

With the highest respect, I wish to share a non-commercial, rights-based initiative aligned with Norway’s leadership in humanitarian protection, disability inclusion, and responsible digital cooperation. Our team has developed an ultra-lightweight artificial intelligence packaged in a single 54 KB HTML file, designed to operate entirely offline and sustain functionality during planned or unplanned disruptions. To lower barriers, repurposed low-power smartphones with small solar chargers can host the tool and be maintained directly by local actors.

Unlike server-dependent AI models, this tool performs all inference locally in the browser. By remaining offline and avoiding network calls, it ensures service continuity, preserves privacy, and reduces cost and operational risk in connectivity-constrained contexts.

The model combines technical minimalism with robust safeguards. It collects no personal data and is accompanied by governance guidance, training outlines, and a template Data Protection Impact Assessment to support adoption by disability organizations, Red Cross and Red Crescent National Societies, and education partners. The principle is clear: effective protection without surveillance.

From inception, the build has adhered to GDPR privacy-by-design and the principles of the UN Convention on the Rights of the Child (UNCRC) and the Convention on the Rights of Persons with Disabilities (CRPD). It aligns with Norway’s advocacy for digital public goods, including its co-founding of the Digital Public Goods Alliance, and with commitments to disability-inclusive development under the Equality for All strategy. It also resonates with Norway’s leadership in the Safe Schools Declaration, underscoring the protection of learning environments in crises. This initiative demonstrates that AI can be inclusive, protective, and rights-based without costly infrastructure.

A functioning version has been tested on repurposed smartphones under minimal-power conditions, with indicative internal benchmarks showing median on-device inference latency below 100 ms for typical queries on entry-level Android devices (for example, Android 8.1 with 1 GB RAM), and approximately one to two percent battery consumption per hour during continuous use. The single-document architecture, with maintained version history, enables transparency, monitoring, and independent review by academic, humanitarian, and regulatory actors. This is an operational model ready for immediate application.

We respectfully propose a modest pilot to demonstrate feasibility and generate field learning, without any financial or operational obligations for the Embassy. If helpful, such a pilot could be conducted with the Norwegian Red Cross or an education partner designated by the Embassy.

If agreeable, I would be grateful for a short virtual demonstration of no more than 15 minutes at a time convenient for the Embassy. I would be happy to adjust to your preferred schedule. Alternatively, a written exchange with a designated focal point is most welcome. Supporting documents — including a one-page overview, a 12-page summary, and a 260-page technical dossier — are available upon request.

Please accept, Your Excellency, the assurances of my highest consideration.

Sincerely,  
 Morgan J.  
 Project Director, AI Necklace for Child Safety Initiative  
 Email: contact@mcorpai.org  
 Website: mcorpai.org

}

[

/\*\*\* Norway — Royal Norwegian Embassies bulk mailer (start today, every 2 hours, max 10/day) \*\*\*/

/\*\*\* Sheet columns: A=Email, B=Status, C=CustomSubject(optional), D=CustomGreeting(optional) \*\*\*/

/\* ========== 0) REQUIRED SETTINGS ========== \*/

const FILE\_ID = '1uTfbMvqRU0SpJvz-SesKg0SyDm66v3y6'; // One-Page 54 KB AI Brief.pdf (in Drive)

const SHEET\_NAME = 'Norway\_recipients'; // Norway sheet tab

const SENDER\_DISPLAY\_NAME = 'M-Corp Ethical AI';

const REPLY\_TO = 'contact@mcorpai.org';

/\* ========== 1) SCHEDULE SETTINGS ========== \*/

const START\_HOUR\_LOCAL = 9; // 기준 시작 시각(스크립트 타임존 기준)

const DAILY\_LIMIT = 10; // 하루 최대 전송 수

const INTERVAL\_HOURS = 2; // 2시간 간격

const SEND\_WINDOW\_MINUTES = 12; // 슬롯 시작 ±12분 허용

const DISPATCHER\_EVERY\_MIN = 5; // 디스패처 주기(분)

/\* ========== 2) CONTENT ========== \*/

const SUBJECT\_VARIANTS = [

'Proposal for Consideration: 54 KB Offline AI for Disability-Inclusive Child Protection',

'Proposal: 54 KB Offline AI for Disability-Inclusive Child Protection',

'Brief Proposal: 54 KB Offline, Rights-Based AI for Inclusive Child Protection'

];

const GREETING\_BASE = 'Your Excellency, Royal Norwegian Embassy';

const VARIANT\_OPENERS = [

'With sincere respect, I am pleased to share an offline, rights-based AI aligned with Norway’s commitments to inclusive protection.',

'Respectfully, I submit for your consideration an entirely offline, privacy-preserving AI consistent with Norway’s leadership in humanitarian protection.',

'I write with highest consideration to introduce a compact, on-device AI designed for low-connectivity environments, in line with Norway’s digital public goods vision.',

'Allow me to present a concise, fully offline AI that supports disability-inclusive protection where networks are unreliable.',

'I wish to briefly introduce an on-device AI approach that advances protection without surveillance, reflecting Norway’s principled stance.'

];

const VARIANT\_CLOSERS = [

'I would gladly adjust to any preferred time or format for an exchange.',

'I remain at your disposal for a short demonstration or written clarification.',

'I would appreciate the opportunity to present a brief demonstration at your convenience.',

'Should it be useful, I can share additional documentation or clarifications.',

'I am grateful for your consideration and would welcome a brief discussion.'

];

const EXEC\_SUMMARY = `Executive Summary:

This initiative introduces a 54 KB offline, rights-based artificial intelligence designed to protect children with disabilities in connectivity-constrained settings. Running fully on-device with no data transmission, it has been validated on low-power smartphones with minimal resource use. The approach aligns with Norway’s leadership in humanitarian protection, disability inclusion, and digital public goods, while placing no financial or operational obligations on the Embassy. Further details and supporting materials are available at the M-Corp AI introductory page (mcorpai.org).`;

const BODY\_MAIN = `

Your Excellency,

Royal Norwegian Embassy,

With the highest respect, I wish to share a non-commercial, rights-based initiative aligned with Norway’s leadership in humanitarian protection, disability inclusion, and responsible digital cooperation. Our team has developed an ultra-lightweight artificial intelligence packaged in a single 54 KB HTML file, designed to operate entirely offline and sustain functionality during planned or unplanned disruptions. To lower barriers, repurposed low-power smartphones with small solar chargers can host the tool and be maintained directly by local actors.

Unlike server-dependent AI models, this tool performs all inference locally in the browser. By remaining offline and avoiding network calls, it ensures service continuity, preserves privacy, and reduces cost and operational risk in connectivity-constrained contexts.

The model combines technical minimalism with robust safeguards. It collects no personal data and is accompanied by governance guidance, training outlines, and a template Data Protection Impact Assessment to support adoption by disability organizations, Red Cross and Red Crescent National Societies, and education partners. The principle is clear: effective protection without surveillance.

From inception, the build has adhered to GDPR privacy-by-design and the principles of the UN Convention on the Rights of the Child (UNCRC) and the Convention on the Rights of Persons with Disabilities (CRPD). It aligns with Norway’s advocacy for digital public goods, including its co-founding of the Digital Public Goods Alliance, and with commitments to disability-inclusive development under the Equality for All strategy. It also resonates with Norway’s leadership in the Safe Schools Declaration, underscoring the protection of learning environments in crises. This initiative demonstrates that AI can be inclusive, protective, and rights-based without costly infrastructure.

A functioning version has been tested on repurposed smartphones under minimal-power conditions, with indicative internal benchmarks showing median on-device inference latency below 100 ms for typical queries on entry-level Android devices (for example, Android 8.1 with 1 GB RAM), and approximately one to two percent battery consumption per hour during continuous use. The single-document architecture, with maintained version history, enables transparency, monitoring, and independent review by academic, humanitarian, and regulatory actors. This is an operational model ready for immediate application.

We respectfully propose a modest pilot to demonstrate feasibility and generate field learning, without any financial or operational obligations for the Embassy. If helpful, such a pilot could be conducted with the Norwegian Red Cross or an education partner designated by the Embassy.

If agreeable, I would be grateful for a short virtual demonstration of no more than 15 minutes at a time convenient for the Embassy. I would be happy to adjust to your preferred schedule. Alternatively, a written exchange with a designated focal point is most welcome. Supporting documents — including a one-page overview, a 12-page summary, and a 260-page technical dossier — are available upon request.

Please accept, Your Excellency, the assurances of my highest consideration.

Sincerely,

Morgan J.

Project Director, AI Necklace for Child Safety Initiative

Email: contact@mcorpai.org

Website: mcorpai.org

Enclosure: One-Page 54 KB AI Brief.pdf`;

/\* ========== 3) UTILITIES ========== \*/

function getSheet\_() {

const sh = SpreadsheetApp.getActive().getSheetByName(SHEET\_NAME);

if (!sh) throw new Error('Sheet "' + SHEET\_NAME + '" not found.');

return sh;

}

function isValidEmail(addr) {

return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(String(addr || '').trim());

}

function hashInt\_(s) {

s = String(s || '');

let h = 0;

for (let i = 0; i < s.length; i++) h = (h \* 31 + s.charCodeAt(i)) >>> 0;

return h >>> 0;

}

function cityFromEmail(email) {

try {

const local = String(email).split('@')[0] || '';

const afterEmb = local.replace(/^emb\./i, '');

const map = {

'the.hague': 'The Hague',

'hague': 'The Hague',

'kuala.lumpur': 'Kuala Lumpur',

'addisabeba': 'Addis Ababa',

'daressalaam': 'Dar es Salaam',

'newdelhi': 'New Delhi',

'buenos.aires': 'Buenos Aires',

'alger': 'Algiers'

};

if (map[afterEmb]) return map[afterEmb];

const parts = afterEmb.split('.').map(s => s.charAt(0).toUpperCase() + s.slice(1));

return parts.join(' ');

} catch (e) {

return '';

}

}

function buildHtmlBody\_(greeting, opener, closer) {

const esc = (s) => String(s || '')

.replace(/&/g,'&amp;').replace(/</g,'&lt;').replace(/>/g,'&gt;');

const g = greeting;

const html = `

<div style="max-width:760px;margin:0 auto;padding:20px 16px;font-family:Arial,Helvetica,sans-serif;font-size:16px;line-height:1.8;color:#111;">

<div style="font-size:19px;font-weight:700;margin:0 0 12px 0;">Proposal for Consideration: 54 KB Offline AI for Disability-Inclusive Child Protection</div>

<p style="margin:0 0 16px 0;">${esc(opener)}</p>

<p style="margin:0 0 16px 0;"><strong>Executive Summary:</strong><br>${esc(EXEC\_SUMMARY.replace(/^Executive Summary:\s\*/,'')).replace(/\n/g,'<br>')}</p>

<p style="margin:0 0 16px 0;">${esc(g)}</p>

${BODY\_MAIN.split('\n').map(p => p.trim() ? `<p style="margin:0 0 16px 0;">${esc(p)}</p>` : '').join('')}

<p style="margin:0 0 16px 0;">${esc(closer)}</p>

</div>`;

return html;

}

function buildPlainText\_(greeting, opener, closer) {

const lines = [];

lines.push('Proposal for Consideration: 54 KB Offline AI for Disability-Inclusive Child Protection');

lines.push('');

lines.push(opener);

lines.push('');

lines.push(EXEC\_SUMMARY);

lines.push('');

lines.push(greeting);

lines.push('');

lines.push(BODY\_MAIN);

lines.push('');

lines.push(closer);

return lines.join('\n');

}

/\* ========== 4) CORE SENDER ========== \*/

/\* Sends exactly one email from the first row with empty Status. \*/

function sendNextOne\_() {

try { scanBouncesAndMark(); } catch (e) { /\* ignore to avoid blocking send \*/ }

const sh = getSheet\_();

const last = sh.getLastRow();

if (last < 2) return false;

const data = sh.getRange(2, 1, last - 1, 4).getValues(); // A..D

let rowIndex = -1, email = '', customSubject = '', customGreeting = '';

for (let i = 0; i < data.length; i++) {

const e = String(data[i][0] || '').trim();

const st = String(data[i][1] || '').trim().toUpperCase();

if (e && st === '') {

rowIndex = i + 2;

email = e;

customSubject = String(data[i][2] || '').trim();

customGreeting = String(data[i][3] || '').trim();

break;

}

}

if (rowIndex === -1) return false; // no pending recipient

if (!isValidEmail(email)) {

sh.getRange(rowIndex, 2).setValue('INVALID');

return false;

}

try {

const h = hashInt\_(email);

const opener = VARIANT\_OPENERS[h % VARIANT\_OPENERS.length];

const closer = VARIANT\_CLOSERS[h % VARIANT\_CLOSERS.length];

const city = cityFromEmail(email);

const greeting = customGreeting || (city ? `${GREETING\_BASE} in ${city}` : GREETING\_BASE);

const subject = customSubject || SUBJECT\_VARIANTS[h % SUBJECT\_VARIANTS.length];

const file = DriveApp.getFileById(FILE\_ID);

const htmlBody = buildHtmlBody\_(greeting, opener, closer);

const textBody = buildPlainText\_(greeting, opener, closer);

GmailApp.sendEmail(email, subject, textBody, {

htmlBody,

attachments: [file.getAs(MimeType.PDF)],

name: SENDER\_DISPLAY\_NAME,

replyTo: REPLY\_TO

});

sh.getRange(rowIndex, 2).setValue('SENT');

Logger.log('Sent to: ' + email);

} catch (err) {

sh.getRange(rowIndex, 2).setValue('ERROR');

Logger.log('Send error for ' + email + ': ' + (err && err.message));

}

return true;

}

/\* ========== 5) DISPATCHER & TRIGGERS ========== \*/

/\* Dispatcher checks the current slot and sends at most one email per slot. \*/

function dispatcher() {

const lock = LockService.getScriptLock();

if (!lock.tryLock(20000)) return;

try {

try { scanBouncesAndMark(); } catch (e) { Logger.log('Bounce scan skipped: ' + (e && e.message)); }

const sp = PropertiesService.getScriptProperties();

const now = new Date();

let campaignStartIso = sp.getProperty('campaignStartIso');

if (!campaignStartIso) return; // must set via create\*()

const start = new Date(campaignStartIso);

const slotLenMin = INTERVAL\_HOURS \* 60;

const diffMin = Math.floor((now.getTime() - start.getTime()) / 60000);

// roll to next day if today’s window passed

const dailyWindowMin = DAILY\_LIMIT \* slotLenMin;

if (diffMin >= dailyWindowMin) {

const next = new Date(start.getTime() + 24 \* 60 \* 60 \* 1000);

sp.setProperties({ campaignStartIso: next.toISOString(), sentCount: '0', lastSentSlot: '-1' }, true);

return;

}

if (diffMin < 0) return; // before start

const slotIndex = Math.floor(diffMin / slotLenMin);

if (slotIndex < 0 || slotIndex >= DAILY\_LIMIT) return;

const lastSentSlot = parseInt(sp.getProperty('lastSentSlot') || '-1', 10);

let sentCount = parseInt(sp.getProperty('sentCount') || '0', 10);

if (sentCount >= DAILY\_LIMIT) return;

const slotStart = new Date(start.getTime() + slotIndex \* slotLenMin \* 60000);

const minutesFromSlot = Math.abs(Math.floor((now.getTime() - slotStart.getTime()) / 60000));

if (minutesFromSlot <= SEND\_WINDOW\_MINUTES && lastSentSlot !== slotIndex) {

const ok = sendNextOne\_();

if (ok) {

sentCount += 1;

sp.setProperties({ sentCount: String(sentCount), lastSentSlot: String(slotIndex) }, true);

}

}

} finally {

try { lock.releaseLock(); } catch (e) {}

}

}

function deleteDispatcherTriggersOnly() {

ScriptApp.getProjectTriggers()

.filter(t => t.getHandlerFunction() === 'dispatcher')

.forEach(t => ScriptApp.deleteTrigger(t));

}

/\* Start today at START\_HOUR\_LOCAL (e.g., 09:00 today) \*/

function createTwoHourCampaignStartingToday() {

deleteDispatcherTriggersOnly();

const now = new Date();

const start = new Date(now.getFullYear(), now.getMonth(), now.getDate(), START\_HOUR\_LOCAL, 0, 0);

PropertiesService.getScriptProperties().setProperties({

campaignStartIso: start.toISOString(),

sentCount: '0',

lastSentSlot: '-1'

}, true);

ScriptApp.newTrigger('dispatcher').timeBased().everyMinutes(DISPATCHER\_EVERY\_MIN).create();

}

/\* Start tomorrow at START\_HOUR\_LOCAL \*/

function createTwoHourCampaignStartingTomorrow() {

deleteDispatcherTriggersOnly();

const now = new Date();

const tomorrow = new Date(now.getFullYear(), now.getMonth(), now.getDate() + 1, START\_HOUR\_LOCAL, 0, 0);

PropertiesService.getScriptProperties().setProperties({

campaignStartIso: tomorrow.toISOString(),

sentCount: '0',

lastSentSlot: '-1'

}, true);

ScriptApp.newTrigger('dispatcher').timeBased().everyMinutes(DISPATCHER\_EVERY\_MIN).create();

}

/\* Start now, aligned to the current 2-hour slot \*/

function createTwoHourCampaignStartingNow() {

deleteDispatcherTriggersOnly();

const now = new Date();

const minutes = now.getHours() \* 60 + now.getMinutes();

const slotLen = INTERVAL\_HOURS \* 60;

const alignedMinutes = Math.floor(minutes / slotLen) \* slotLen;

const start = new Date(now.getFullYear(), now.getMonth(), now.getDate(), Math.floor(alignedMinutes / 60), alignedMinutes % 60, 0);

PropertiesService.getScriptProperties().setProperties({

campaignStartIso: start.toISOString(),

sentCount: '0',

lastSentSlot: '-1'

}, true);

ScriptApp.newTrigger('dispatcher').timeBased().everyMinutes(DISPATCHER\_EVERY\_MIN).create();

}

/\* ========== 6) BOUNCE SCAN ========== \*/

function scanBouncesAndMark() {

const sh = getSheet\_();

const last = sh.getLastRow();

if (last < 2) return;

const rows = sh.getRange(2, 1, last - 1, 2).getValues();

const indexByEmail = {};

for (let i = 0; i < rows.length; i++) {

const email = String(rows[i][0] || '').trim().toLowerCase();

if (email) indexByEmail[email] = i + 2;

}

const query = 'newer\_than:7d (from:mailer-daemon OR subject:"Delivery Status Notification" OR subject:"Undelivered Mail Returned to Sender")';

const threads = GmailApp.search(query, 0, 200);

const seen = {};

threads.forEach(th => {

th.getMessages().forEach(msg => {

const body = (msg.getBody() || '') + '\n' + (msg.getPlainBody() || '');

const candidates = [];

const m1 = body.match(/Final-Recipient:\s\*rfc822;\s\*([^\s<>]+)/ig) || [];

m1.forEach(line => {

const hit = (line.split(/rfc822;/i)[1] || '').trim();

if (hit) candidates.push(hit);

});

const m2 = body.match(/[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}/ig) || [];

m2.forEach(e => candidates.push(e));

candidates.forEach(raw => {

const email = String(raw || '').replace(/[<>\s,;]+/g, '').toLowerCase();

const row = indexByEmail[email];

if (row && !seen[email]) {

const cur = String(sh.getRange(row, 2).getValue() || '').trim().toUpperCase();

if (cur === '' || cur === 'SENT' || cur === 'ERROR') {

sh.getRange(row, 2).setValue('BOUNCED');

}

seen[email] = true;

}

});

});

});

}

/\* Daily 04:00 bounce scan \*/

function createBounceScanDailyTrigger() {

ScriptApp.newTrigger('scanBouncesAndMark').timeBased().atHour(4).everyDays(1).create();

}

/\* Every 15 minutes bounce scan (near-real-time exclusion) \*/

function createBounceScanEvery15Min() {

ScriptApp.newTrigger('scanBouncesAndMark').timeBased().everyMinutes(15).create();

}

function deleteAllTriggers() {

ScriptApp.getProjectTriggers().forEach(t => ScriptApp.deleteTrigger(t));

}

/\* ========== 7) MANUAL TESTS ========== \*/

/\* 테스트 1통(카운트 미포함) \*/

function sendOneNowForTest() {

const lock = LockService.getScriptLock();

if (!lock.tryLock(10000)) return;

try { sendNextOne\_(); }

finally { try { lock.releaseLock(); } catch (e) {} }

}

/\* 테스트 1통(카운트 포함) \*/

function sendOneNowForTestCounted() {

const lock = LockService.getScriptLock();

if (!lock.tryLock(10000)) return;

try {

if (sendNextOne\_()) {

const sp = PropertiesService.getScriptProperties();

const c = parseInt(sp.getProperty('sentCount') || '0', 10);

sp.setProperty('sentCount', String(c + 1));

}

} finally { try { lock.releaseLock(); } catch (e) {} }

}

]

<

1. 가나 주재 노르웨이 대사관 emb.accra@mfa.no
2. 그리스 주재 노르웨이 대사관 emb.athens@mfa.no
3. 나이지리아 주재 노르웨이 대사관 emb.abuja@mfa.no
4. 남아프리카공화국 주재 노르웨이 대사관 emb.pretoria@mfa.no
5. 네덜란드 주재 노르웨이 대사관 emb.hague@mfa.no
6. 네팔 주재 노르웨이 대사관 emb.kathmandu@mfa.no
7. 대한민국 주재 노르웨이 대사관 emb.seoul@mfa.no
8. 덴마크 주재 노르웨이 대사관 emb.copenhagen@mfa.no
9. 러시아 주재 노르웨이 대사관 emb.moscow@mfa.no
10. 루마니아 주재 노르웨이 대사관 emb.bucharest@mfa.no
11. 리투아니아 주재 노르웨이 대사관 emb.vilnius@mfa.no
12. 말라위 주재 노르웨이 대사관 emb.lilongwe@mfa.no
13. 말레이시아 주재 노르웨이 대사관 emb.kuala.lumpur@mfa.no
14. 멕시코 주재 노르웨이 대사관 emb.mexico@mfa.no
15. 모로코 주재 노르웨이 대사관 emb.rabat@mfa.no
16. 모잠비크 주재 노르웨이 대사관 emb.maputo@mfa.no
17. 미국 주재 노르웨이 대사관 emb.washington@mfa.no
18. 미얀마 주재 노르웨이 대사관 emb.yangon@mfa.no
19. 사우디아라비아 주재 노르웨이 대사관 emb.riyadh@mfa.no
20. 방글라데시 주재 노르웨이 대사관 emb.dhaka@mfa.no
21. 베트남 주재 노르웨이 대사관 emb.hanoi@mfa.no
22. 벨기에 주재 노르웨이 대사관 emb.brussels@mfa.no
23. 불가리아 주재 노르웨이 대사관 emb.sofia@mfa.no
24. 브라질 주재 노르웨이 대사관 emb.brasilia@mfa.no
25. 세르비아 주재 노르웨이 대사관 emb.belgrade@mfa.no
26. 스웨덴 주재 노르웨이 대사관 emb.stockholm@mfa.no
27. 스위스 주재 노르웨이 대사관 emb.bern@mfa.no
28. 스페인 주재 노르웨이 대사관 emb.madrid@mfa.no
29. 슬로바키아 주재 노르웨이 대사관 emb.bratislava@mfa.no
30. 싱가포르 주재 노르웨이 대사관 emb.singapore@mfa.no
31. 아랍에미리트 주재 노르웨이 대사관 emb.abudhabi@mfa.no
32. 아르헨티나 주재 노르웨이 대사관 emb.buenos.aires@mfa.no
33. 아이슬란드 주재 노르웨이 대사관 emb.reykjavik@mfa.no
34. 아일랜드 주재 노르웨이 대사관 emb.dublin@mfa.no
35. 에스토니아 주재 노르웨이 대사관 emb.tallinn@mfa.no
36. 에티오피아 주재 노르웨이 대사관 emb.addisabeba@mfa.no
37. 영국 주재 노르웨이 대사관 emb.london@mfa.no
38. 오스트리아 주재 노르웨이 대사관 emb.vienna@mfa.no
39. 요르дан 주재 노르웨이 대사관 emb.amman@mfa.no
40. 우크라이나 주재 노르웨이 대사관 emb.kyiv@mfa.no
41. 이스라엘 주재 노르웨이 대사관 emb.telaviv@mfa.no
42. 이집트 주재 노르웨이 대사관 emb.cairo@mfa.no
43. 이탈리아 주재 노르웨이 대사관 emb.rome@mfa.no
44. 인도 주재 노르웨이 대사관 emb.newdelhi@mfa.no
45. 인도네시아 주재 노르웨이 대사관 emb.jakarta@mfa.no
46. 일본 주재 노르웨이 대사관 emb.tokyo@mfa.no
47. 중국 주재 노르웨이 대사관 emb.beijing@mfa.no
48. 체코 주재 노르웨이 대사관 emb.prague@mfa.no
49. 칠레 주재 노르웨이 대사관 emb.santiago@mfa.no
50. 태국 주재 노르웨이 대사관 emb.bangkok@mfa.no
51. 캐나다 주재 노르웨이 대사관 emb.ottawa@mfa.no
52. 케냐 주재 노르웨이 대사관 emb.nairobi@mfa.no
53. 콜롬비아 주재 노르웨이 대사관 emb.bogota@mfa.no
54. 쿠바 주재 노르웨이 대사관 emb.havana@mfa.no
55. 크로아티아 주재 노르웨이 대사관 emb.zagreb@mfa.no
56. 탄자니아 주재 노르웨이 대사관 emb.daressalaam@mfa.no
57. 터키 주재 노르웨이 대사관 emb.ankara@mfa.no
58. 튀니지 주재 노르웨이 대사관 emb.alger@mfa.no
59. 페루 주재 노르웨이 대사관 emb.lima@mfa.no
60. 포르투갈 주재 노르웨이 대사관 emb.lisbon@mfa.no
61. 폴란드 주재 노르웨이 대사관 emb.warsaw@mfa.no
62. 프랑스 주재 노르웨이 대사관 emb.paris@mfa.no
63. 핀란드 주재 노르웨이 대사관 emb.helsinki@mfa.no
64. 필리핀 주재 노르웨이 대사관 emb.manila@mfa.no
65. 헝가리 주재 노르웨이 대사관 emb.budapest@mfa.no
66. 호주 주재 노르웨이 대사관 emb.canberra@mfa.no
67. 레바논 주재 노르웨이 대사관 emb.beirut@mfa.no
68. 라트비아 주재 노르웨이 대사관 emb.riga@mfa.no
69. 세네갈 주재 노르웨이 대사관 emb.dakar@mfa.no  
    >